WHOI-86-3

A Miniature Deep Sea Temperature Data Recorder: Design, Construction, and Use

by

R. Koehler and R. P. von Herzen

Woods Hole Oceanographic Institution
Woods Hole, Massachusetts 02543

January 1986

Technical Report

Funding was provided by the National Science Foundation under grant Nos. OCE 82-14658 and OCE 83-00073. Additional support was provided by U.S. Geological Survey of Woods Hole to begin development of instrumentation; and to the Ocean Industry Program of the Woods Hole Oceanographic Institution to complete the development.

Approved for Distribution:

Robert C. Spindel, Chairman
Department of Ocean Engineering

TABLE OF CONTENTS

List of Figures

List of Figures for the Appendices

ABSTRACT

A miniature temperature recorder has been developed to be used with the hydraulic piston sediment corer (HPC) on the Deep Sea Drilling Project (DSDP). The instrumentation fits into pressure-sealed slots in the wall of the HPC, allowing temperature measurements to be made simultaneously with coring operations. Temperatures from -2 to 70°C are measured to a resolution of about 0.01°C. Up to 1300 13-bit measurements are recorded in random access memory (RAM), at a sampling rate ranging between 0.1 s to over 100 min., as specified by the operator in a program loaded into a microprocessor of the instrument. During recording the instrumentation uses about 3.5 mamp at 7.5 volts, which can be supplied for about 20 hours of operation by a custom-made pack of silver-oxide batteries. The corer is normally left motionless in the sediment for about 10 min. to allow extrapolation of the measured temperatures to equilibrium *in-situ* temperature. Examples of data from DSDP Leg 86 are given.

I. INTRODUCTION

Marine geothermal measurements require that the vertical temperature gradients in sediments be determined accurately. This is usually accomplished by instrumentation lowered to the sea floor from an oceanographic ship or drilling platform. Probes and cores used with oceanographic vessels are typically designed to penetrate the bottom from 1 to 10 m, with the gradient measured by 4-7 thermal probes rigidly attached along the probe. In deep-sea drilling, instrumentation has been used to measure temperatures at intervals during drilling frequently to depths of 200-300 m (Erickson et al., 1975; Yokota et al., 1980), and occasionally to 600 m below the sea floor (Erickson and Von Herzen, 1978).

The much greater penetration achieved by drilling techniques provide the opportunity to investigate variability of heat flow with depth. Although many results indicate relatively constant heat flow with depth (Erickson et al., 1975; Hyndman et al., 1985), as expected for steady-state conductive thermal transfer, oceanographic studies suggest that some regions may have more complicated thermal signatures. On or near continental margins, processes such as rapid sedimentation or slumping, or bottom water temperature variations, may produce more complex or disturbed thermal profiles. Hydrothermal circulation in ocean crust is a process which dominates heat transfer over much of the youngest sea floor, and extends to sea floor as old as 80 m.y. in some ocean basins (Anderson et al., 1977). Recently it has been found that this mechanism may involve slow fluid transfer through the overlying sediments (Anderson et al., 1979; Von Herzen et al., 1982; Becker and Von Herzen, 1982). The vertical component of interstitial fluid flow will produce a distinctive temperature profile with depth different from conductive profiles, essentially a exponential curve with a depth scale which depends primarily on the rate of vertical flow. Even extremely slow rates of pore fluid flow ($\sim 10^{-8}$cm sec^{-1} = 0.3cm/yr) have important ramifications for schemes to dispose of wastes in the sea floor, esp. radioactive wastes which must be isolated for 10^5 yrs or more.

Successful in-situ measurements of temperature in Deep Sea Drilling Project (DSDP) holes began with Leg 5 in 1969. Over the past decade, measurements have been obtained at only a few depths downhole in a small fraction of the total number of holes drilled by DSDP, due to several factors:

1) Drilling objectives did not justify the additional time required by such measurements.

2) The harsh environment of the drilling operations, primarily large shocks and accelerations, rendered downhole instrumentation inoperative (this factor has been substantially reduced by development of all solid-state instrumentation (e.g. Yokota et al., 1980)).

3) Sediment physical properties did not allow measurement of equilibrium temperatures (either too soft to hold the measurement probe steady, or too hard to penetrate).

4) Sediments are thermally disturbed due to drilling.

The successful development of a DSDP hydraulic piston corer (HPC) has allowed retrieval of remarkably undisturbed sediment cores to depths of 200 to 300 m beneath the sea floor. The HPC is a nearly ideal vehicle for an in-situ temperature sensor because it deploys well below (up to 10 m) the drilled and thermally disturbed hole for core retrieval (Fig. 1). Furthermore, its almost continuous deployment for the upper several hundred meters of sediments on many DSDP holes provides the opportunity to obtain detailed depth profiles of temperature with little or no cost in additional ship time.

Here we describe, in detail, instrumentation developed to obtain in-situ temperature during deployments of the HPC, along with examples of data. The instrumentation was first used on DSDP Leg 86 in 1982 (Horai and Von Herzen, 1985) and extended through the drilling program carried out on D/V GLOMAR CHALLENGER. Its use presently (January 1986) continues with the Ocean Drilling Program from D/V JOIDES RESOLUTION.

II. DESIGN CONSIDERATIONS

The HPC (hydraulic piston corer) is up to 10 m long and about 9 cm diameter (Anonymous, 1984). The most useful location of the temperature sensor is in the cutting shoe (the tip of the coring tool) where it is farthest from the temperature disturbances of the drill bit. Also, the tip of the coring tool is thinner than the rest of the core barrel, which suggests that it takes less time to reach equilibrium temperature with the sediment. The temperature recorder electronics could be anywhere, as long as it did not impede the operation of the coring tool.

Initially, we considered putting the electronics in a hollow (doughnut shaped) cylinder at the top of the coring tool, and running the sensor wires down the length of the coring tool. Although the users were willing to give up 15 cm of coring length for the electronics at the top of the core barrel, we could not determine how to run the sensor wires past the threaded sections of the coring tool to the sensor in the tip.

The cutting shoe has a 1.43 cm thick wall. If a portion of that wall could be hollowed out to make room for the electronics, then the sensor wires could be connected directly to the temperature recorder. If the electronics were contained on a hybrid circuit, they could be put into a space no larger than an ordinary large integrated circuit package. The $8000 for the hybrid circuit layout and masks plus $1800 for each unit produced were reasonable costs.

Engineers at DSDP designed as large a cavity as possible in the coring shoe to hold a very small electronics package and withstand the 10,000 psi pressure at the bottom of the 21,000 ft drill string. The modified coring shoe evolved in steps. Initially, the cavity was to be covered by a lid on the outside of the coring tool, but the possibility existed for loosening of the cap and jamming inside the drill string. The second version had the top and the bottom of the coring shoe separate, exposing the cavity for the electronics. The two sections were bolted together and O-ring seals kept the water and pressure out of the cavity. The strength of the bolts was

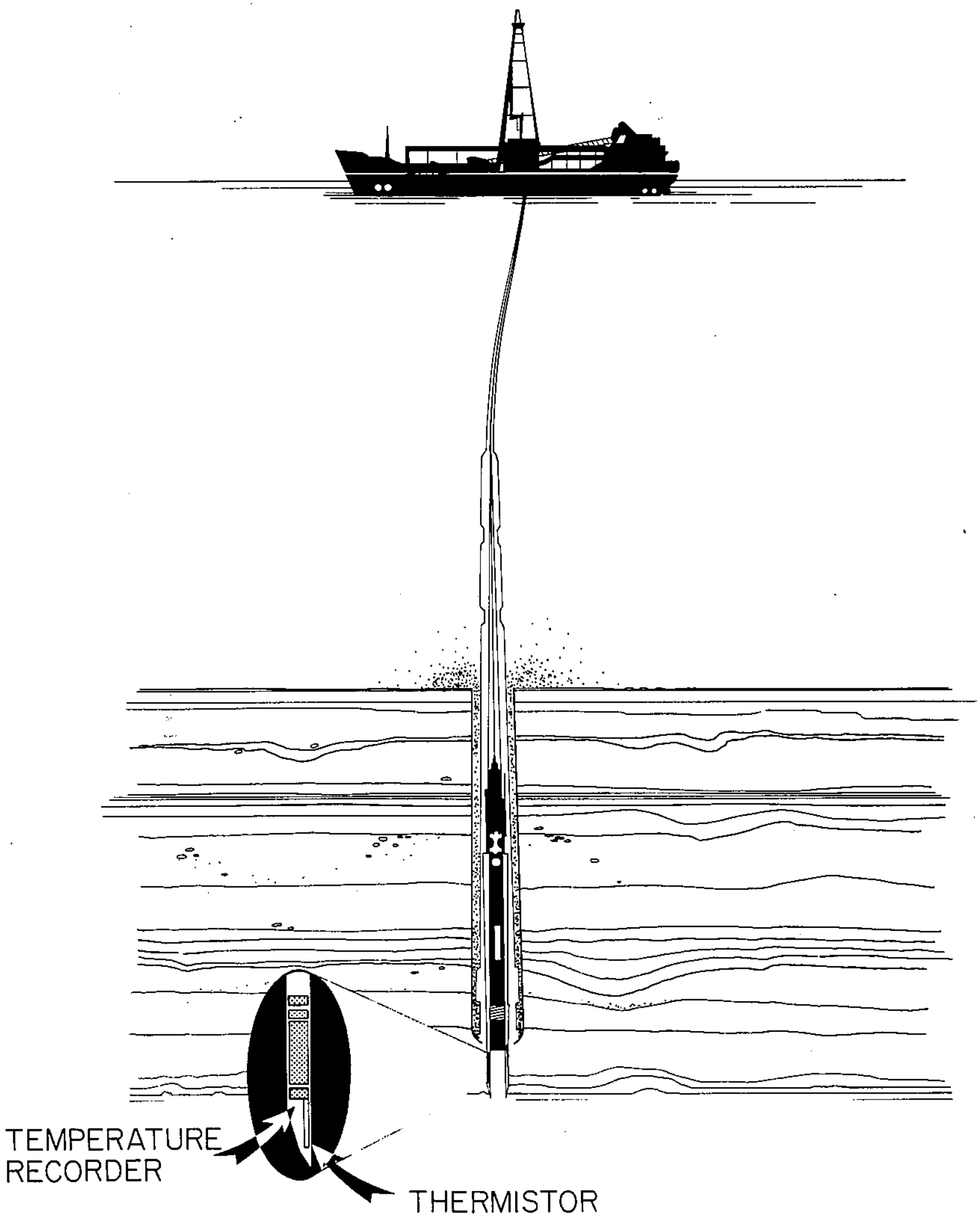

Figure 1. Schematic configuration of drill ship deploying HPC during coring operations

determined as insufficient to withstand the side forces the coring shoe could encounter. The third version replaced the bolted fastening with a threaded section. The strongest heat treatable steel available was used for the shoe, so that the cavity could be as wide as possible.(See Appendix A for the calculation of the strength of the cavity walls.)

The instrumentation was designed to record temperatures up to 70°C. It also had to survive the shocks encountered during the traverses of the drill string at high speed (200 m/min), which we estimated at approximately 5000 g (equivalent to a free-fall of 5 ft. stopped in 0.01 in.). The hybrid could withstand these shocks, and the rest of the printed circuit board was potted to keep the parts from moving and breaking off the printed circuit board.

III. THE INSTRUMENT

A. DESCRIPTION

The temperature recorder can make and store 1300 temperature measurements and occupies 16 cm^3 (1 $in.^3$). The battery was designed to last from 35 to 50 hours (active, the lesser amount when cold) to 400 hours (data stored mode). It occupies only 10 cm^3 (0.6 $in.^3$) (see Fig. 2). Initially, the recorder is coupled to a small computer and its program is loaded each time before use. Up to 8 time delays and recording rates can be inserted, although only one is normally used.

After the program is loaded, and the recording parameters selected, the two halves of the coring shoe are screwed together (Fig. 3). The electronics and battery package are sealed from ocean pressures and water by O-ring seals. The coring shoe is taken to the drilling platform where it is attached to the bottom of the hydraulically driven piston corer. A tilt switch within the battery package activates the timing of the recorder when the shoe and coring tool are lifted to vertical (with the cutting edge pointing down).

The corer is lowered to the bottom of the drill string, driven into the sediment, and held there for about 5 to 10 minutes. The corer is pulled to the top of the drill string and the core and core shoe are removed. In the lab the coring shoe is opened, and a plug from the computer connected to the temperature recorder. The data are read into the computer over an RS-232C (teletype) line. The computer stores the 1300 temperature points on tape or a disk. The data, which are in raw form, can be converted to actual temperatures and stored or plotted. Also, an algorithm to determine the in situ bottom core temperature can be applied, by extrapolating the asymptotic behavior of the temperature measurements. A personal computer used to communicate with the temperature recorder is less expensive than constructing a special unit to store and process the data.

B. DESIGN

The temperature recorder block diagram is shown in Figure 4. A thermistor in a Wheatstone bridge provides voltage for an analog-to-digital

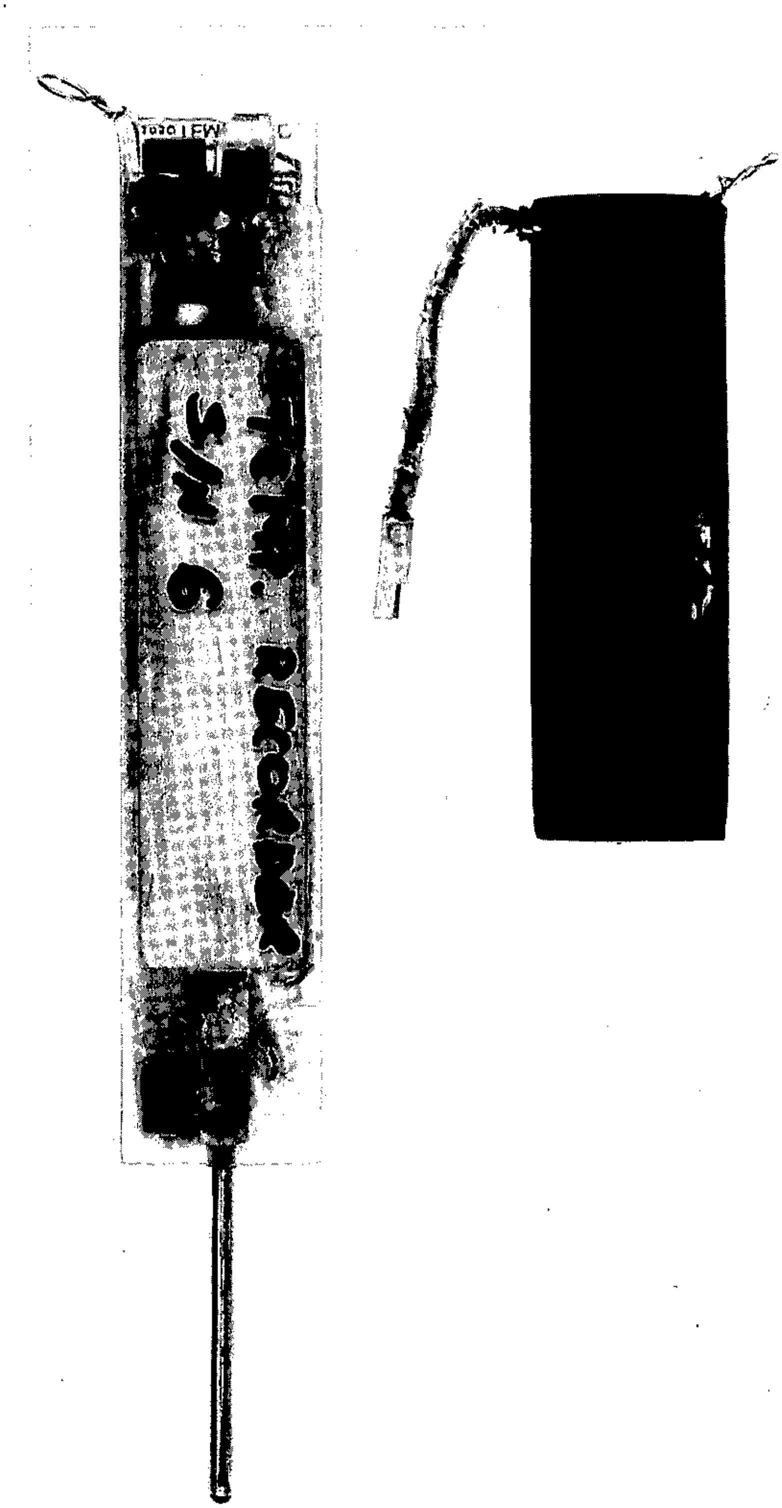

Figure 2. Photograph of temperature recorder and battery. The body of the recorder is 12 cm long.

Figure 3. Photograph of instrumentation and battery pack outside of special hydraulic piston coring (HPC) cutting shoe (2 pieces) designed to contain them.

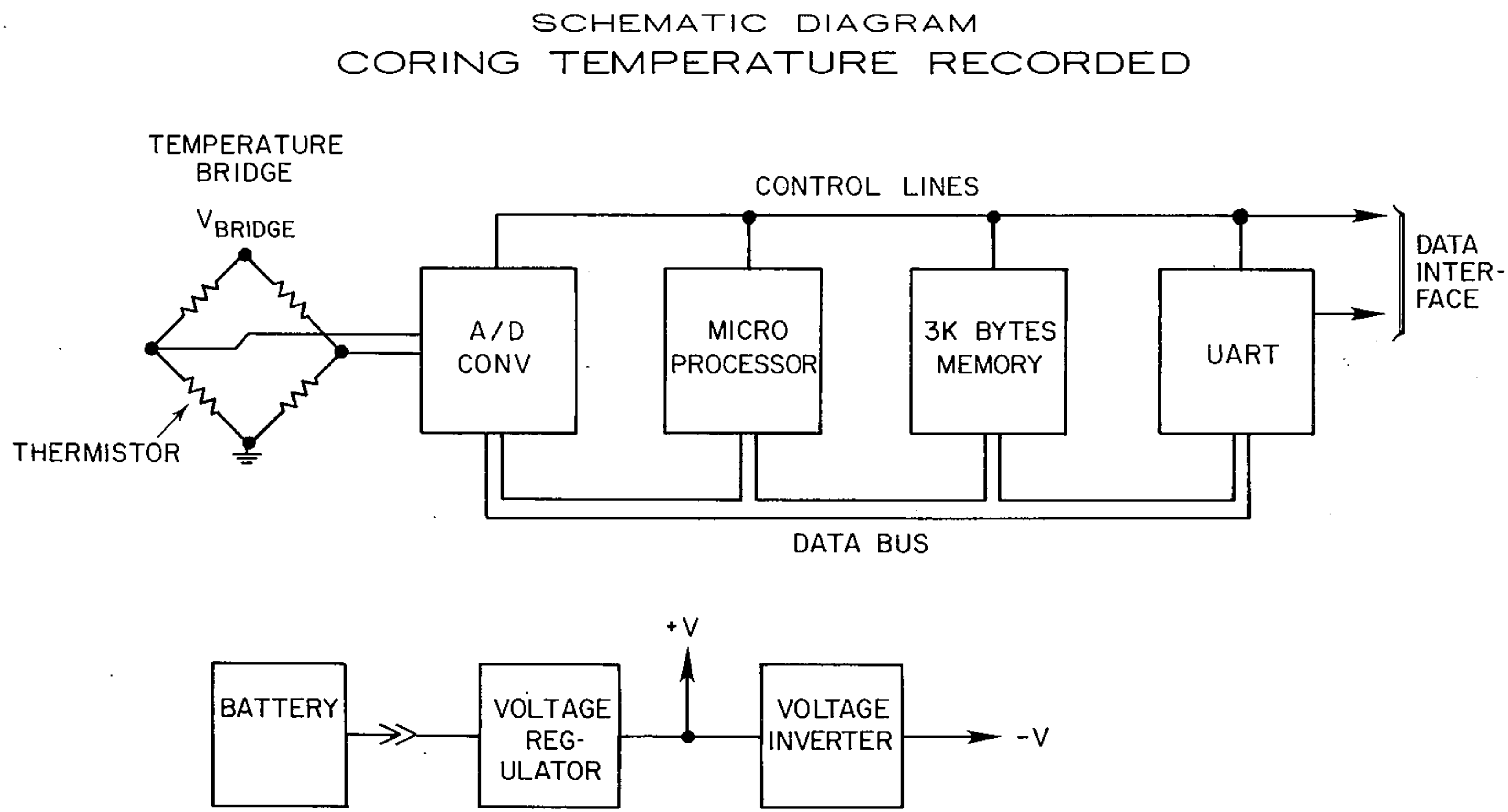

Figure 4. Block diagram of HPC temperature recorder.

(A/D) converter. An RCA 1802 microprocessor is used to couple the output of the A/D converter to random access memory (RAM). The microprocessor also controls when the A/D converter takes a measurement. There are 3k bytes of memory; about 0.5k are used for the program and the rest is used to store data. A universal asynchronous receiver transmitter (UART) is used to couple the computer to a teletype type output and input.

The input instructions are transferred through the UART to the computer. The program and answer to inquiries are transferred from the computer through the UART to the microprocessor and its memory.

A voltage regulator provides the +5 volts for the digital and analog circuitry. A voltage inverter provides -5 volts used only by the A/D converter. Five silver oxide hearing aid batteries provide 8 volts to the voltage regulator. To save energy, the voltage to the A/D converter is switched on only when it is being used. To save additional energy, when the data memory is filled, the voltage is dropped to 4 volts, and the microprocessor halts.

After the recorder is brought back to the surface, the coring shoe is opened up in the lab to gain access to the recorder electronics. The recorder is plugged into the on-deck computer, the recorder voltage is raised to 5 volts and the microprocessor on the recorder is started up again. The internal program then dumps the data out, stops the microprocessor and lowers the recorder voltage to 4 volts when completed. Because of the low standby power drain, the battery does not have to be disconnected. In operation at sea, the entire unit has to be removed each time so that the O-ring seals and the threaded opening to the electronics can be cleaned of mud, grease, and any dirt which may have accumulated during the deployment.

C. CONSTRUCTION

The circuitry had to be made quite small in order to fit into the small space available. A hybrid circuit is very small in size, and can be made relatively economically in small quantities. The first question for a hybrid circuit is determining whether it will be a thick or thin film hybrid. All hybrids are made on a ceramic substrate, typically .025 in. thick. A thin film hybrid is a single layer of gold lines placed on the ceramic through a mask. Since such a technique does not allow for wiring cross-overs, the circuit cannot be very complex. A thick film hybrid consists of alternate layers of gold lines and glass insulator on a ceramic substrate. Each layer is silk screened on and the pattern is fired before the next layer is applied. Interconnections between the wires on different layers are made through holes ("vias") in the insulating layers. Because of the multiple layers, multiple cross-overs are possible to accommodate a more complex circuit. The complexity of this circuit, which used 20 integrated circuits, required a thick film hybrid circuit.

The second problem was finding a hybrid manufacturer who was willing to manufacture only a few (5-10) units. There was a fairly large engineering effort in laying out the circuit, and sources for just a few of each type of integrated circuit had to be found. Larger hybrid companies preferred to use their limited engineering expertise on products with a larger production. Even the smaller company we found (Transistor Specialities, Inc, Danvers, Mass.) accepted the job with reluctance, but then was determined to do the job, even though it was more complicated (5 metalization layers) than his colleagues in other companies thought prudent.

The design at first incorporated two 40-pin packages (0.6 in. spacing between pin rows, 2 in. long) to hold the circuitry. Later the package manufacturer came out with a 3-in.-long package with pins on 0.6 in. spacing. It was long enough to hold the whole circuit, except for certain large parts. This simplified the printed circuit, since no connections would now be needed between two hybrid packages. We were fortunate that someone else underwrote the tooling for this larger package, as a die designed for a new package costs around $14K. Additional costs would be incurred for the lid die and the solder preform die.

The circuit had to be designed for integrated circuits that are available in dice form. Not only does the manufacturer need to have them available in dice form, but the dice distributer must be willing to stock the part. They are not very interested if the part is not in great enough use to allow them to sell the minimum order from the manufacturer. For RCA this minimum is one wafer of dice. For others it might be 100 pieces, even though the piece costs $100 each. Japanese manufacturers did not seem to be interested in selling dice at all. Buying small numbers of dice is a problem. The designer must consider what is available in small numbers, or be willing to purchase more dice than needed. We even designed for one part that later was discontinued at the time of purchase. Fortunately, the hybrid had been inadvertently layed out for a larger equivalent part that was still available. Also, lead times were long, and slower moving items are not manufactured frequently.

After the hybrid is manufactured, it must be tested without physically disturbing the circuit very much. A single bond wire can be gently raised from the substrate with only a modest chance of ruining the IC; the entire IC can be replaced with a greater chance of ruining the whole hybrid. Generally only the hybrid manufacturer can electrically probe the hybrid. It was not difficult to determine if the circuit was working incorrectly. Ingenuity was required to devise tests to determine what part was at fault, without removing any of the parts.

Ultimately, we solved all the problems, but it was considerably more difficult than working with a printed circuit. The circuit should be designed so that faults can be isolated. The most difficult problem is designing a technique for determining which of the several ICs fastened to the same lead is at fault.

D. PRINTED CIRCUIT LAYOUT

We wanted to put all the components inside the hybrid in order to minimize the size of the package. However, a few parts were left out of the hybrid either because they were too large to fit, there was insufficient space in the hybrid package for them, or we wanted to be able to change them

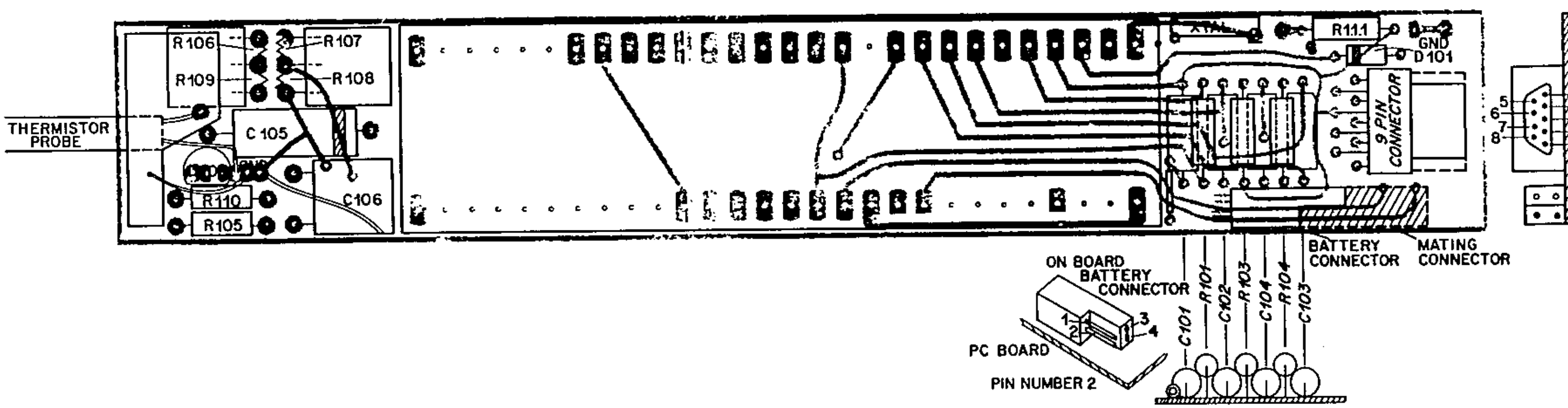

Figure 5. Configuration of printed circuit board and components used in HPC temperature instrumentation.

after the hybrid was built. Means were needed to connect the hybrid pins to these parts, to the temperature sensor and to the connectors for power and communication. The usual method for interconnecting parts, a printed circuit, was used (Fig. 5). It was unusual in that it was very thin, only 1/64 in. thick, in order to use as little space as possible.

The bridge components and the A/D integrating and zero offset capacitors were placed at one end of the printed circuit. The hybrid package was placed in the middle. A thin (.008 in.) insulating sheet of fiberglass-epoxy board was placed between the metal hybrid package and the printed circuit. On the other end were the tantalum bypass capacitors, crystal, voltage regulator resistors, A/D frequency resistor, the data/control connector, and the battery connector.

Initially, the circuit runs were wide (0.32 in.) and the spaces between them thin (.010 in.), with the idea that the lines could be robust and the spaces between the lines did not have to be very strong. This spacing proved to be too close. The etching was not complete in several places, and fine hairs of solder from the edges of the traces could easily bridge the gaps between the lines. After an initial run of several instruments, the circuit board was layed out again to accommodate some changes, and the opportunity was used to make the traces and spaces between them equal. The board was changed to accomodate some smaller packaged bridge resistors (two to a package instead of one), an external voltage regulator for the bridge, and centering of both the large integrating capacitor and the communication connector. The centering made use of the larger clearance available in the center of the kidney-shaped hole into which the package was placed (Fig. 6). Initially, it was supposed that the thin board would bend, and the mold would force the taller components at the edge of the board inside the confines of the space available. The narrow board was quite stiff, and would not bend much. This stiffness was actually a blessing, because this board was all that really provided any stiffness to the package. This stiff board made an especially good method to secure the connector to the rest of the circuitry without flexing its soldered connections.

All parts leads were bent over parallel to the board to maximize packing efficiency. Even so, some of the leads were .025 in. thick, about 10% of the available height. The bending of leads also provided for a stronger connection, since the board was so very thin.

E. POTTING

Very large shocks to the electronics package were anticipated as it travelled down the drill pipe. Similar problems had been solved by enclosing everything in a semi-rigid potting material. The rigidity of the potting material kept the parts from being ripped from place by their own inertia during shock. Some flexibility is desirable to prevent the parts from being broken or leads sheared from the slight differences in thermal expansion between the potting material and the items being potted. A reported high failure rate of hermetically sealed tantalum capacitors of the 150D series caused us to examine more closely several of these capacitors.

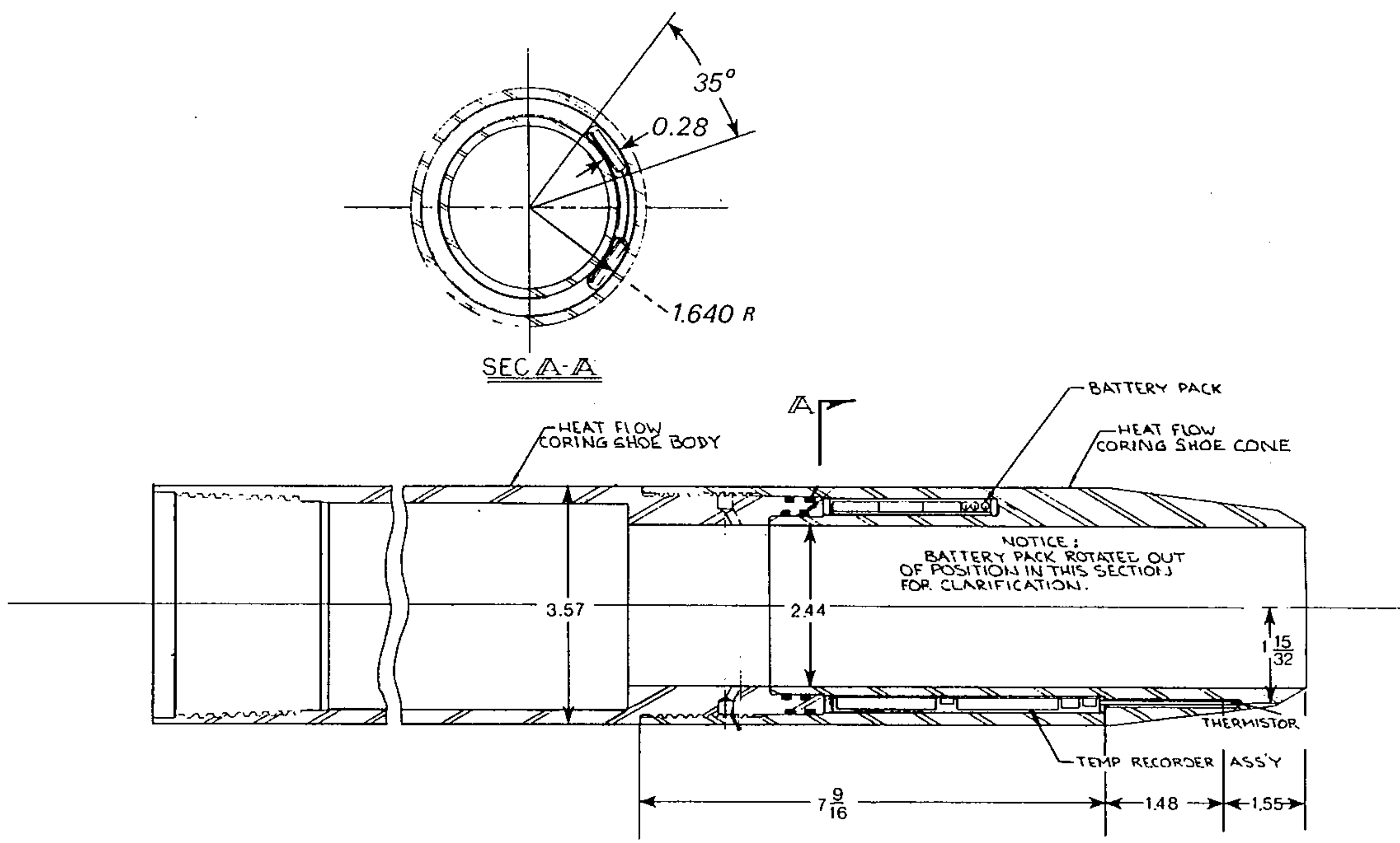

Figure 6. Sectional drawing of temperature instrumentation deployed in slot in wall of HPC. Dimensions given in inches.

They appeared fragile with a gossamer appearance, so solid tantalum capacitors potted in epoxy were used.

Several potting materials were tried. A very hard epoxy chipped and flaked off in the thin sections, although it proved satisfactory for the battery packs, where the epoxy sections were thicker. All the slightly flexible epoxies and polyurethanes became softer and much more flexible at 70°C. A polyurethane from CONAP (Olean, N.Y.), EN-9, was selected for this service. It too became quite flexible at 70°C, but it remained tough.

Creating a suitable mold was one of the most difficult tasks. In an attempt to save money, we tried to take an impression of the cavity into which the electronics were to fit. We got a rubber impression, but it was flexible. Eventually, we got a hard epoxy impression, from which a hard polyurethane mold was made. Several electronic packages were cast which had to be sanded and filed to enable them to fit into the cavity.

Later, we received a sample cavity made of steel from the DSDP in which our molded units could be tested for fit. This test fixture was converted into a two-piece mold by cutting it in half, and replacing all but .010 inches of the cut with a shim. This solved our molding problems. A hard mold made by a mold-making shop would have cost $1,200, which in retrospect would have been less expensive than our mistakes.

The instrument casting was made by filling the mold from the bottom, with the mold in a vertical position, which reduced the possibility of entrapped air. Because of the thin covering and bubble-prone potting techniques, the whole circuit was thinly coated with a polyurethane material (Laminar X500; Midland Div., The Dexter Corp., Hayward, CA) designed for that purpose. This would keep the circuit dry, protecting it from condensation as the package cooled off, and from accidental sea water exposure while opening up the package to read out the data. Another problem was solvent evaporating from the circuit board coating. The drying time recommended by the manufacturer of the printed circuit board coating was long enough for the coating to harden, but not long enough for all the solvent to have evaporated. The first board coating that we used worked, but we changed in mid-stream to a second type, which had the extra solvent.

The thermistor used to measure temperatures is Part No. GB42JM63 (Fenwall Electronics, Framingham, MA) with a nominal resistance of 15k ohms at 25°C. It is potted within a small (0.120 in. O.D.) tube with a high thermal conductivity silicone rubber (Eccosil 4952, Emerson and Cumming, Canton, MA). This probe is then potted in proper position with the electronics package to comprise a rigid unit.

The battery potting is done with a hard epoxy, (Stycast 2651-MM, Emerson and Cumming Co., Canton, MA), as the batteries could withstand the shrinkage forces of the setting epoxy. We first used a fairly hard rubber mold, filled from the top. A short cable with a small connector on it is used to power the electronics module. The main problem is keeping the epoxy from running out along the cable, making it stiff. The cable is given an accordian fold, to accomodate slight differences in cable length. The lack of stiffness in the mold was compensated for by making the package slightly thinner than the cavity. Thus clearance could exist where the battery protruded slightly into the rubber mold. Later, the steel mold for the electronics was plugged to a length of 3 in., and used to mold the batteries.

F. BATTERIES

A cavity for the batteries in the coring shoe was placed circumferentially, rather than axially, adjacent to the electronics cavity, to minimize the length of the coring shoe. The corer takes a 10 m long (maximum) core and we did not want to use any more space than necessary. A cavity with the same cross section as for the electronics was used for the battery, even though it could have been a little narrower. The 3 in. length accomodated the 5 button cells. The Eveready S76E silver oxide button cell for hearing aids was chosen for the largest cell that would fit into the space available. The cell is 0.455 in. dia. x 0.211 in. thick. A battery assembly company (Brentronics, Commack, NY) welded 5 of these together for us, with the cells side by side.

The cells have 190 mamp-hr capacity at 25°C, and can operate at 70°C if not held at that temperature for too long. The electronics uses only 3.5 mamp maximum when the A/D converter is running, which should give 50 hours of life. This is more than sufficient for perhaps 20 each or more deployments to the bottom for coring and return (average time 2 hrs. each). The electronics package is opened up each time, and the battery pack can be easily replaced at the end of its life. Even longer life is achieved in the "computer running only" state with a current drain of 1.3 mamp for 140 hours (about 6 days), or in the standby mode (memory retained, computer idling) with a current of 0.4 mamp for a 475 hour life (20 days).

The operating time with the A/D running, based on a 190 mamp-hr battery capacity at 25°C, is reduced to about 35 hrs (70% capacity) at 0°C. However, the battery pack was exhausted after only 20 hours at 0°C, with a recorder current measured at 3.5 mamp. This was verified in the laboratory by connecting a resistor to the battery pack which would draw 3.5 mamp at 7.5 volts (the nominal battery voltage, which is nearly constant during discharge). The additional reduction to 20 hours is unexplained. It could be caused by old batteries, which have reduced life, or by the heat from welding tabs to such small cells. Fortunately, 20 hours is still sufficient for 8 to 10 measurements.

The battery pack includes an inexpensive tilt sensor which commences timing when the corer is lifted to the vertical position. Originally it had been placed in the electronics package, but there was insufficient space for it, and it had to be put into the battery pack.

A small 4-pin connector was used to connect the battery pack to the electronics. The cable part was fastened to the disposable battery, to eliminate any cable fatigue in the permanent electronics.

IV. CALIBRATION

A. TEMPERATURE CALIBRATION

The temperature recorders are calibrated in a stable temperature controlled bath from 0 to 60°C at each 10°C intervals. The bath temperature is measured with a platinum resistance thermometer and bridge while the temperature recorder output is monitored with a computer. The resistance of the thermistor in the temperature recorder at each temperature point is calculated and used to fit to the thermistor equation (see below). The parameters of the thermistor equation for a particular recorder are used for calibration of the data acquired by that particular temperature recorder.

1. TEMPERATURE BATH TECHNIQUE

A stable temperature bath is used to control the temperature of the temperature recorders. We used a Tronac (Orem, Utah) bath, with a stability of about 2 m°C or better. Each temperature recorder is put into the finger of a waterproof surgeons glove and connected to a cable for plugging into the interface box. The glove and recorders are taped to a rigid plastic bar and inserted as far into the bath as the top of the glove will safely allow. A plastic bar is used because of its low heat conductivity and heat capacity. A separate ground wire is placed in the bath and connected to the ground connection (black binding post) on the interface box. This shorts out the 50 volts ac that would otherwise be in the bath and couple noise into the recorder. We calibrate all the recorders that are on hand at one time, to save the time it takes to run through the temperature range with the bath. Each recorder cable is plugged into the interface box at each temperature. Although we have not used it, a switch box would be desirable.

We started at the lowest temperature (0°C) and worked up, because the bath can be heated more quickly than it can be cooled. The bath temperature was measured with a laboratory standard platinum resistance thermometer and a precision resistance bridge. Calibration of the platinum resistance thermometer is maintained in the usual way with a triple point of water cell at 0.010°C. The two 10 turn potentiometers (coarse and fine) used to set the bath temperature on the controller have been replaced with a stable resistance box, to enable return to a given temperature setting.

The interface box was connected to an HP 85 desktop computer which both down-loaded the RCA 1802 A/D Read program into the recorder and read the recorder output and listed it on the screen. The HP 85 program, which read the recorder output, also had an option to average 50 readings, to get a better estimate of the measurement, which usually jumped between 2 and occasionally 3 least significant bits. The output of the interface box is in hexadecimal numbers in ASCII on an RS232-C line, so that any computer can be used.

2. LINEARITY AND SENSITIVITY

The readings from the recorder are used to determine the resistance of the thermistor at each temperature calibration point by using the

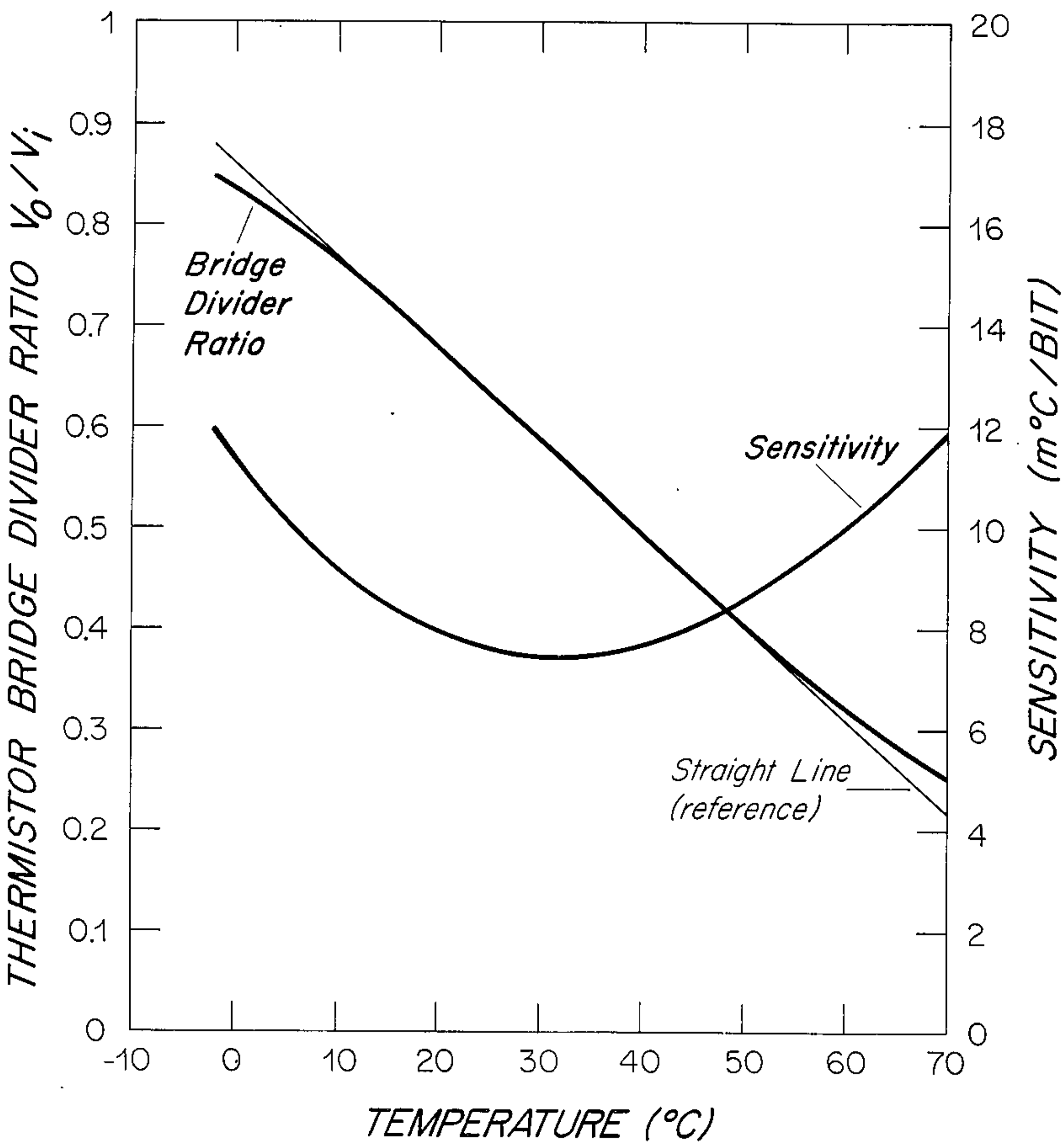

Figure 7. Thermistor bridge divider ratio and sensitivity vs. temperature.

parameters of the .01% accuracy resistors used in the bridge circuit for the thermistor (Fig. D-1, Appendix D). The output of a resistance bridge with a thermistor in one leg is fairly linear in temperature (see Fig. 7). This linearity keeps the sensitivity fairly uniform over the range, varying from 8 to 11 m°C/bit for this recorder range of -2 to 70°C. These resistance values vs. temperature are used in a least squares best fit to the Steinhart and Hart equation for thermistors. The parameters of this thermistor equation are then used to convert the raw data measured by the recorder into temperature. The thermistors used in the recorders are matched to within 0.25°C over the 0 to 70°C temperature range. To optimize temperature accuracy, the thermistor calibration coefficients for each individual recorder must be used in data conversion.

The temperature recorders should be recalibrated often enough to keep any changes from affecting the measurements. Some calibration shifts of up to 50 m°C were found in 2 instruments after one year (see below). The problem was that we do not know when the calibration changed, so we do not know which data were affected. Hence, a method to check the calibration at sea is desirable. In these cases, only the calibration offset changed, not the slope, so perhaps a single point calibration would be sufficient while at sea to assure the validity of the data.

B. THERMISTOR CURVE FITTING TO THERMISTOR CALIBRATIONS

We have found that the thermistor calibrations are fit very well (to better than a few millidegrees over a 60° range) by a 3 parameter equation (Steinhart and Hart, 1968)

$$T^{-1} = A + B \ln R + C (\ln R)^3$$

where T = temperature (K)

R = thermistor resistance (ohms)

A, B, C are constants, different for each thermistor. The constants A, B, C are determined from a least squares best fit to thermistor calibration data as explained in greater detail in Appendix E.

C. THERMAL RESPONSE TESTS OF SPECIAL HPC CORING SHOE (DSDP)

The following describe tests of the thermal response of the shoe, as made in the laboratory at WHOI (January 1982) using various materials and procedures. A calibrated thermistor was placed in a 1/8 in. outer-diam. steel tube, which was in turn inserted in the mating hole in the shoe designed to contain the temperature sensor. The thermistor was located about 1-1/2 in. from the cutting edge (we did not attempt to vary this distance during these tests). The thermistor was connected to a signal processor and data logger (actually a breadboard of the circuitry employed in the in-situ recording system).

Thermal responses were obtained in 3 different materials:

1) in a well-stirred water bath
2) in gelatin
3) in clay (potter's clay).

All of these tests were carried out in a bucket of the respective materials about 10 in. diam. x 12 in. deep. The shoe was inserted into the middle of the materials starting at time $t = 0$. The results in Fig. 8 shows the fraction (F) remaining of the total temperature change for each experiment, which is calculated as $F = \frac{T_\infty - T(t)}{T_\infty - T_o}$

where T_o = HPC shoe temperature before starting test

T_∞ = temperature of medium (or bath) in which HPC shoe is inserted

$T(t)$ = HPC shoe temperature at time t after start of test.

The curves in Fig. 8 show that the HPC shoe will decay to about 20-25% of its initial temperature anomaly 5 minutes after penetration. An exact determination of in-situ temperatures requires fitting of the thermal response curve to a theoretical (numerical) model (see Sec. VI.). The thermal response curves have some peculiarities which may be explained by the experimental conditions:

1. The thermal decay curve in the well-stirred water bath has an inflection between 20-30 sec when plotted on a logarithmic time scale. This may represent the thermal response of the shoe geometry. On the other hand, it may be the effect of the test conditions. The water outside of the HPC cutter is well-stirred whereas the water inside the shoe was mixed only by raising and lowering the shoe a few times during the test. This was probably first done at about 20-30 sec. after beginning this test.

2. The thermal response of the HPC in gelatin appears slightly faster than in clay, whereas the thermal conductivity of the gelatin (1.48 $mcal\,°C^{-1}cm^{-1}sec^{-1}$, or TCU) is less than half that of the clay (3.65 TCU). We surmise that there may have been a thin layer of air between the HPC cutter and the clay, caused by the way in which the shoe was forced into the clay. The relatively high shear strength of the clay caused the level of clay inside the shoe to be much less than the level outside (3-7/8 in. vs. 7-3/4 in., respectively, from the shoe cutting edge) during the test, whereas these level differences during the tests with gelatin were observed to be much less. (This problem would not be expected during normal use of the HPC in-situ due to the high hydrostatic pressure.)

Another possibility is that T_∞ in the gelatin (18.8°C) was measured incorrectly. It was significantly lower than that of the clay even though all tests were done in the same laboratory. It was noted after the tests that the gelatin rapidly disappeared as a result of dehydration (evaporation), perhaps causing the surface of the gelatin to have a lower temperature than the interior of the bucket used in the tests.

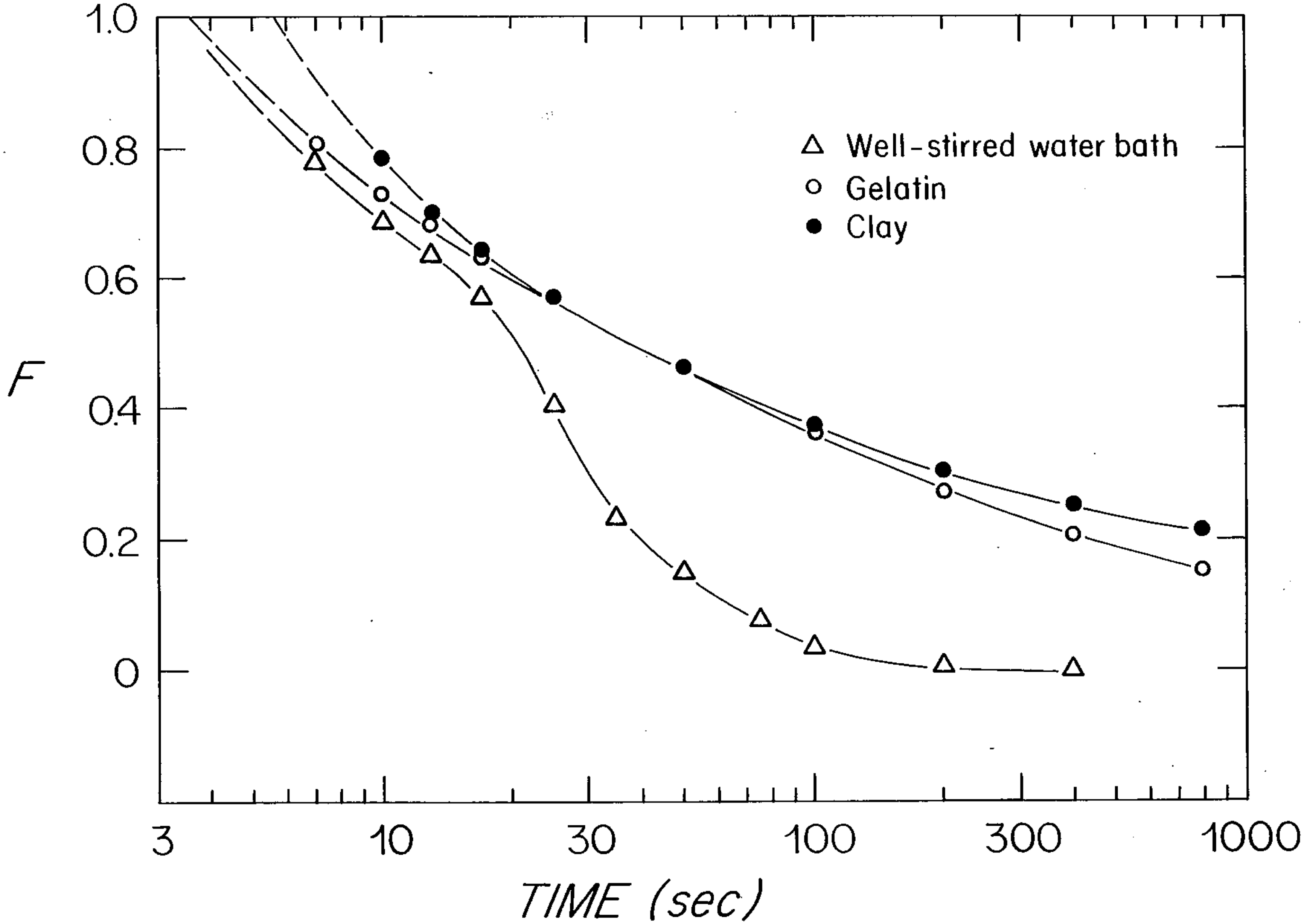

Figure 8. Thermal response of the instrumentation within the HPC shoe in laboratory tests with different types of materials.

V. PROGRAM LOADING AND DATA HANDLING

The temperature recorder, contained within the wall of the coring shoe, can make up to about 1300 temperature measurements during coring operations, consisting of: 1) the descent of the coring tool to the bottom of the drill hole, 2) the insertion of the coring tool in the sediment (the temperature measurements that we want), and 3) the ascent of the coring tool to the surface. Temperature can be measured over the range from -2°C to +70°C, with an average resolution of 9 m°C. A time delay can be set so that the temperature recorder is near the bottom of the drill string before recording begins. The sample interval is used to spread out the 1300 measurements to bracket the time that the coring tool is in the sediment. If necessary, more than one sample interval can be used, with the faster rate in the center of the expected time in the sediment and slower rates before and after, in case the center time does not coincide with the time in the sediment. However, there is no time recorded, so that the timing of data acquisition rate changes must be deduced from the programming and accurate counting of the data.

A. INTERFACE BOX

The temperature recorder communicates through its UART with a desktop computer interfaced through the computer's RS-232 communication port. However, the UART voltages from the temperature recorder and the RS-232 voltages from the computer are different. So an interface box was designed to change the voltage levels between the temperature recorder and the interface box. The box and its cables also provide a means to connect from the subminiature 9 pin connector on the temperature recorder to the larger 25 pin RS-232 connector on the computer.

The interface box also has a manually controlled RUN/RESET/LOAD function, which controls the program load operation from the computer to the temperature recorder. The RUN function is used to communicate with the temperature recorder. The RESET function resets the microprocessor in the recorder, and resets its program counter to 0. The LOAD function enables the recorder to load successive bytes into its memory from the RS-232 line of the desktop computer. This is how the program is loaded into the temperature recorder. When the program has run and data is to be read out, the RESET function is used to set the program counter to 0, and the data is transmitted when returned to the RUN function. This data is read by the desktop computer.

Either the PRO-350 or HP-85 computers can be selected. An HP-85 computer was used during instrument development because it was available. The HP-85 has program control of several other lines in the RS-232 connector, and these lines were programmed to control the load and reset functions that have to be operated manually with the PRO-350. Therefore, there are level changing circuits in the interface box for the reset, load, and power down/power up lines.

The circuitry of the interface box changes the +12 volts and -12 volts signals on the RS-232 line from the computer to the +5 volts and 0 volts signals of the UART in the temperature recorder. The table below describes the signals.

Table 1. RS-232 Interface Box Connections

SIGNAL	INTERFACE BOX		RECORDER
	RS-232 CONNECTOR	RECORDER CONNECTOR	RECORDER CONNECTOR
Data from desktop computer to temperature recorder	3	3	2
Data from temperature recorder to desktop computer	2	2	1
Signal ground	7	1	0
+5 volts from temperature recorder	-	7	6
+Battery	-	6	5
Reset temp. recorder computer	5 *	8	7
Load temp. recorder computer	8 *	9	8
Power up (same signal as reset)	5 *	5	4
Power down	6 *	4	3

* Connected only when the HP-85 is selected.

The data rate is 1200 baud, determined by the temperature recorder. The RS-232 data lines use negative true logic, whereas the control lines use positive true. The recorder uses positive true logic for both data and control lines.

B. DESCRIPTION OF COMPUTER PROGRAMS.

There are two types of computer programs. One type (suffix .ASC) is run by the temperature recorder, and controls the A/D converter, sample interval, storage of data, and reading data out. These programs are written in assembly language for the RCA 1802 microprocessor in the temperature recorder.

The other type of programs (suffix .BAS) are run by the DEC PRO-350, to load the 1802 programs into the temperature recorder, provide a terminal to interact with the temperature recorder's request for data collection parameters, and read the data from the temperature recorder, and convert the raw data to temperature. These programs can be run on other small computers by making suitable changes in the programs to accomodate the pecularities of the particular computer. The programs are written in BASIC for the DEC PRO-350.

The PRO-350 must load the program into the recorder each time the recorder is used. The recorder has only RAM (no ROM), so the program disappears each time the power is disconnected from it. Also, part of the recording program is written over by data, when that part of the program is no longer needed, to make room for more data.

The PRO-350 programs are listed first, along with the corresponding menu item that selects them. All the programs are provided on a floppy disk, but are run from the hard disk because they need it to run during their "interim storage" phase, which is used to make room for storage of data from the recorder.

PRO 350 PROGRAMS	CORING TOOL TEMPERATURE RECORDER (MENU)
PROBASIC.BAS	
MENU.BAS	
LOADER.BAS	
	1. LOAD PROGRAM
LOAD1.BAS	2. LOAD PARAMETERS
LOAD2.BAS	3. DUMP DATA
LOAD3.BAS	4. DISPLAY DATA FILE
LOAD3A.BAS	5. PRINT FILE DATA
LOAD4.BAS	6. CONVERT AND PRINT FILE DATA
	7. END

PROBASIC.BAS - A short instruction program for running the MENU program.

MENU.BAS - A program to select thermal conductivity, downhole instrument operations, or exit.

LOADER.BAS - The main operating program for downhole operations. It contains the operation menu which has 7 entries (the menu is called CORING TOOL TEMPERATURE RECORDER). The load program sub-menu and program for loading the xxxx.ASC programs into the temperature recorder is the first item in the menu.

Menu items 2 through 6 call the sub-programs LOAD1.BAS through LOAD4.BAS respectively and menu item 7 returns to the MENU.BAS program. The following sub-programs return to LOADER.BAS when finished.

LOAD1.BAS - to load operational parameters into the temperature recorder and start it running.

LOAD2.BAS - dumps data from the temperature recorder into the computer, display it to the screen, and store data to disk.

LOAD3.BAS - displays raw data for screening.

LOAD3A.BAS - prints raw data on a printer for hard copy.

LOAD4.BAS - converts raw data files to resistance and temperature files (xxxx.TXT). This program includes all the temperature recorder calibration data. The program converts the raw data, saves results to disk, and provides a hard copy on the printer.

The temperature recorder (1802) programs are loaded into the recorder by selecting item 1, LOAD PROGRAM, from the CORING TOOL TEMPERATURE RECORDER menu. The programs that can be loaded, with their corresponding menu selection and a brief description are listed below.

RECORDER PROGRAMS	LOAD PROGRAM (MENU)
MEMTS2.ASC	1. LOAD MEMORY TEST
TREC.ASC	2. LOAD MAIN PROGRAM
ADTEST.ASC	3. LOAD A/D TEST
AVERG.ASC	4. LOAD AVERAGING PROGRAM
PLYBK.ASC	5. LOAD PLAYBACK PROGRAM

MEMTS2.ASC - Used to test memory of the temperature recorder.

TREC.ASC - The main program for data acquisition by the temperature recorder.

ADTEST.ASC - This program makes A/D readings of the temperature bridge and transmits them through the UART to the desk top computer several times a second.

AVERG.ASC - A version of TREC2.ASC which averages 8 temperature measurements by the A/D and stores the average.

PLYBK.ASC - This program is used to read out data from the recorder when the readout procedure built into TREC2.ASC fails.

The use of these programs is described in the "Temperature Recorder Operating Manual".

VI. PERFORMANCE

A. EXPERIENCE IN USING THE HPC TEMPERATURE INSTRUMENT

The first and most extensive field use of the HPC instrument to date (Jan. 1986) in the Deep Sea Drilling Project was on Leg 86 in the Western Pacific (Horai and Von Herzen, 1985). The relatively large data recovery from that leg, approximately 40 downhole measurements at six different sites, was primarily due to the dedication of K. Horai and the DSDP staff. After the first one or two sites, during which experience was being accumulated, one person (K. Horai) could maintain the instrumentation and process the data at a rate of one measurement on every other core (approximately one measurement each 3 to 4 hours). K. Horai (pers. comm., 1984) indicates that under normal operating conditions, with at least two instruments available for alternate use and preparation, one person per watch dedicated to the program could probably obtain measurements on every core (each 1-1/2 to 2 hours) during continuous operations.

Some problems were encountered on the initial use of the instrument. The difficulties with potting materials and techniques described above led to the instruments on Leg 86 being successfully potted only immediately before they were taken to sea. These potted units did not quite fit into the slots in the core nose for which they were made, so that K. Horai spent several hours on each unit to trim (sand) them down to size. We are not certain why these potted units did not fit with the same mold which was successfully used to pot trial units in the laboratory. One possibility is that inclusion of the instrument within the potting reduces the amount of shrinkage because there was less potting material to shrink compared to the trial units.

Another problem on Leg 86 was that the programming of a delay time in the instruments before taking data never appeared to perform properly. Apparently (K. Horai, pers. comm., 1984) the setting of a time delay in the program TREC (see appendix) resulted in arbitrary start times of the instrument, different than set up in the program. All of the successful runs on Leg 86 were made with the delay time set equal to zero, a condition which always started data acquisition soon after picking up the core barrel and initiation by the tilt switch. The failure with start times other than zero may be a software fault or operational confusion, as instruments tested in the laboratory appeared to perform properly.

During measurements at the first site of Leg 86 (site 576), it was noted that small oscillations of temperature over the period of HPC extension (approximately 5 to 10 minutes) were observed, apparently due to vertical heave oscillations of the ship (Fig. 9). It was observed that release of tension in the cable used to lower the HPC, as well as lowering the drill string 1-2 m, resulted in much smoother thermal decay curves (Fig. 10). Apparently the cable tension is responsible for transmitting motion to the HPC when it is extended into the bottom. Also, some data appeared to be repeated over short intervals (3-4 data points), most likely a software fault in the processing of data. This problem occurred less frequently at subsequent sites.

Other characteristics of the measurements important for obtaining accurate in-situ values were observed on Leg 86 operations as well as subsequent legs. Although, in some cases, the measurement was disturbed after the initial penetration, probably as a result of drill ship motion, sometimes a good thermal decay was observed after a few minutes of disturbance (Fig. 11). It appears that the smooth thermal decay after the disturbances can be used to obtain accurate in-situ values, using numerical methods described below. Another type of measurement is illustrated in Figure 12. Here the temperature appears to decay smoothly after the initial frictional penetration pulse, but subsequently increases irregularly thereafter. This behavior typically occurred for measurements attempted in the shallower portions of holes ($\leq$100 m), and it seems probable that the irregular temperature increases are caused by slow settling of the corer and drill string in the soft sediments. If a sufficiently long ($\geq$2 min) undisturbed record of the thermal decay exists after penetration, reasonably accurate in-situ temperature may be obtained.

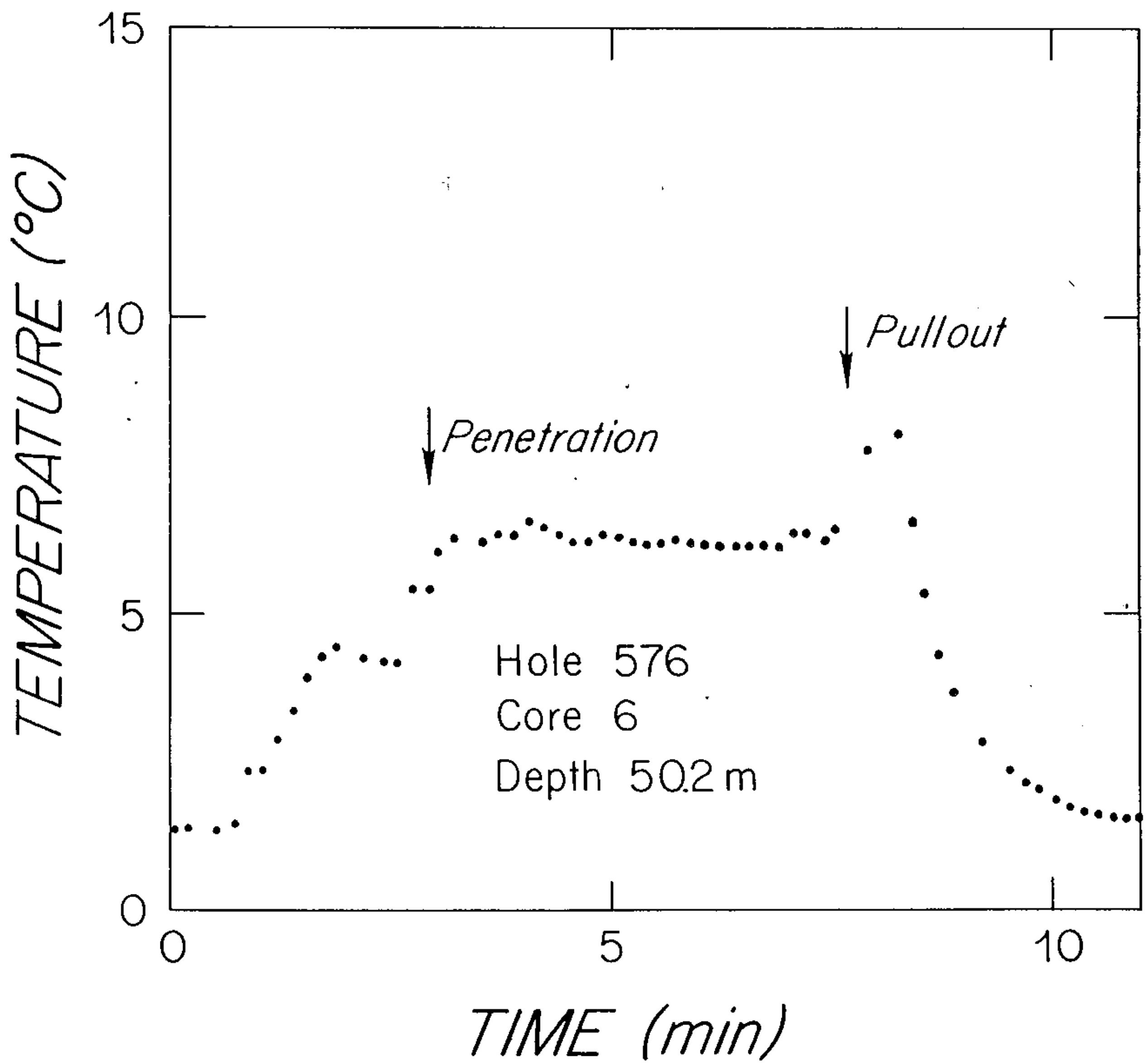

Figure 9. In-situ temperature measurement on Leg 86, site 576, core 6. Note small oscillations of temperature after penetration.

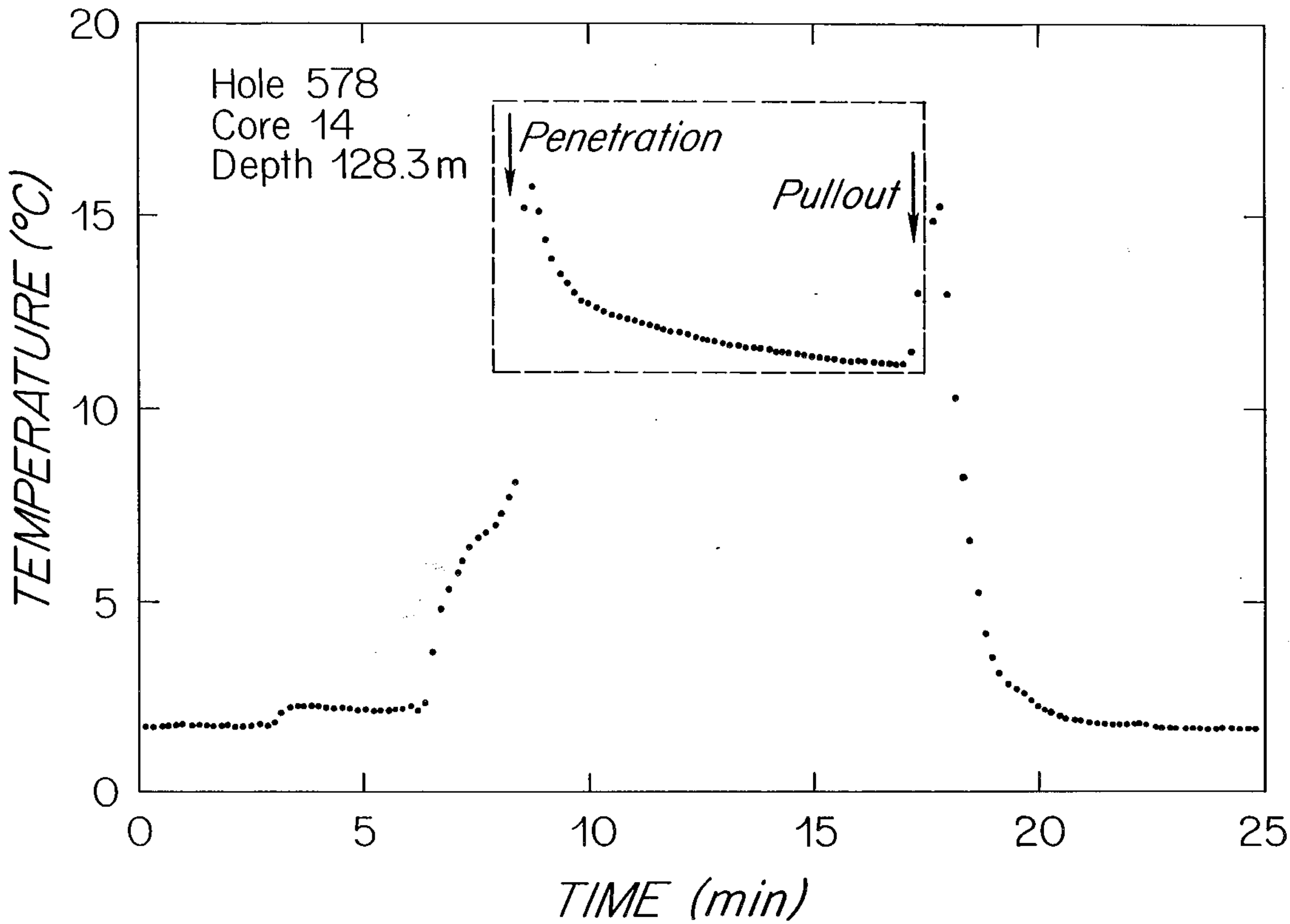

Figure 10. In-situ temperature measurements on Leg 86, site 578, core 14. Note relatively smooth thermal decay after penetration. Outlined portion of record expanded in Fig. 13.

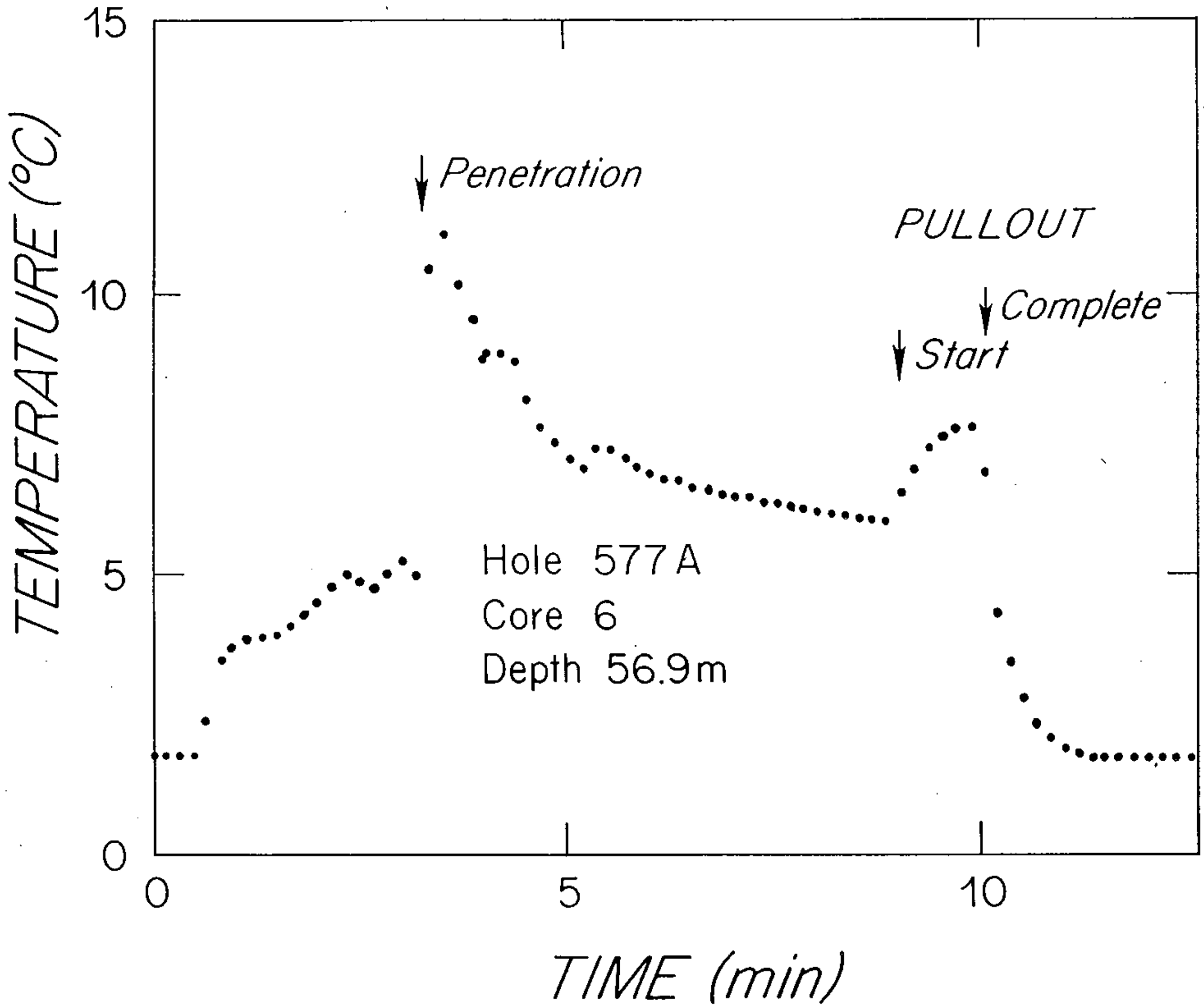

Figure 11. In-situ temperature measurements on Leg 86, site 577A, core 6. Note relatively disturbed record over about 2 min. after penetration then a smooth decay.

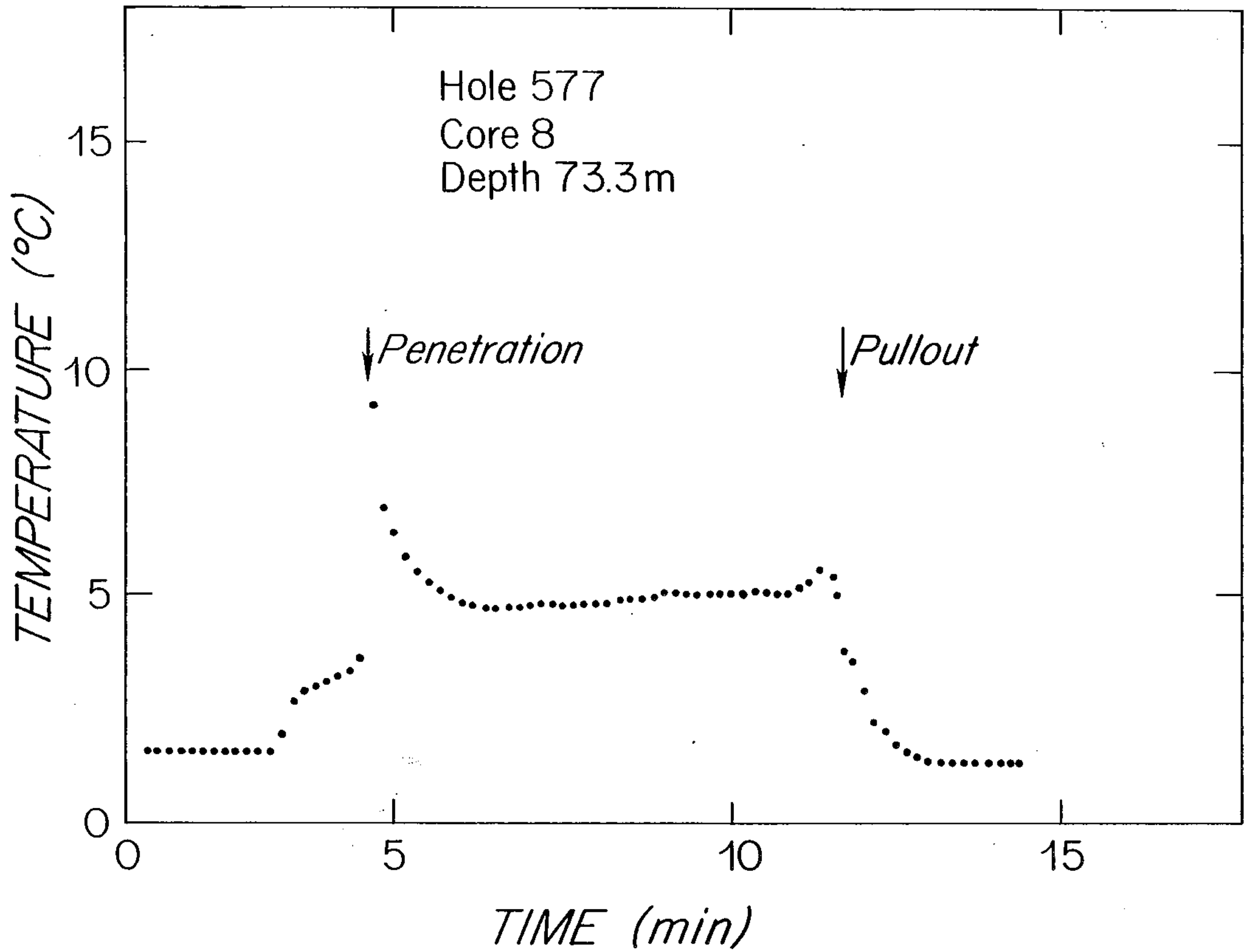

Figure 12. In-situ temperature measurements on Leg 86, site 577, core 8. Note sequential decrease, then a gradual increase, of temperatures after penetration.

The HPC temperature instrumentation was also used on several legs subsequent to Leg 86, but not as extensively. Unfortunately, three sets of instrumentation were lost on subsequent legs (87, 89 and 90), which significantly limited the amount of data obtained. According to DSDP engineers (Dave Huey, pers. comm., 1984) these losses were caused by failure of the HPC hardware, and had little to do with the operation of the instrumentation itself. Although measurements had been planned for Leg 96, none were attempted because of the unstable sediments and the fear that the leaving the HPC in the bottom without circulation, even for a few minutes, might stick the drill pipe (W. Bryant, pers. comm.). Water circulation is accomplished by pumping seawater through the drill pipe from the ship, allowing it to flow out the drill bit and up around the annulus between the drill pipe and the hole walls. This cannot be accomplished with the HPC extending out the bit. A useful development would allow seawater circulation through ports in the drill string above the drill bit after extension of the HPC. This might allow the HPC to remain extended at the same time that circulation above the HPC would prevent any cave-in or packing of sediments around the drill string.

Overall, it seems that the primary factor limiting use of the HPC is the lack of interest or motivation of shipboard scientists to obtain data. Otherwise there is little reason why such measurements could not be made an almost routine part of HPC operations.

B. EXTRAPOLATION TO EQUILIBRIUM TEMPERATURES

During the time the HPC is in the bottom, the frictional heat generated by the penetration is conducted away to the surrounding sediments (Fig. 10). The two-dimensional radial conduction theory for the physical configuration of the HPC has been solved numerically by K. Horai (pers. comm., 1984). Both the theory and experimental field data show that the excess temperature of the HPC over in-situ values is reduced to about 15-20% of its initial value after about 10 minutes in the sediment. It is not practical to leave the extended core barrel in the sediment much longer than this, so that an accurate extrapolation of the frictional heating pulse is necessary.

A comparison between the theory of Horai and an actual measurement is shown in Figure 13. The agreement between theory and practice is extremely close after the first few data points (40-60 sec.). The lack of agreement during the first few data points occurs when the temperature distribution in the HPC wall is probably not uniform. A small remaining problem seems to be that the theoretical temperatures fall slightly below the temperatures measured near the end of the thermal decay. This may be due to the fact that the anomalous heat in the HPC is being conducted axially, as well as radially. The excess heat in the HPC is conducted preferentially toward the thin end of the wedge shaped cutting shoe, where the thermistor is located (Fig. 6). This would tend to cool the HPC more rapidly than if the HPC were infinitely extended along its axis, as assumed in the two-dimensional theory.

Therefore, in the measurements for Leg 86, Horai and Von Herzen (1985) assume that the true in-situ equilibrium temperature lies between the last measured point during the HPC penetration, and the theoretical

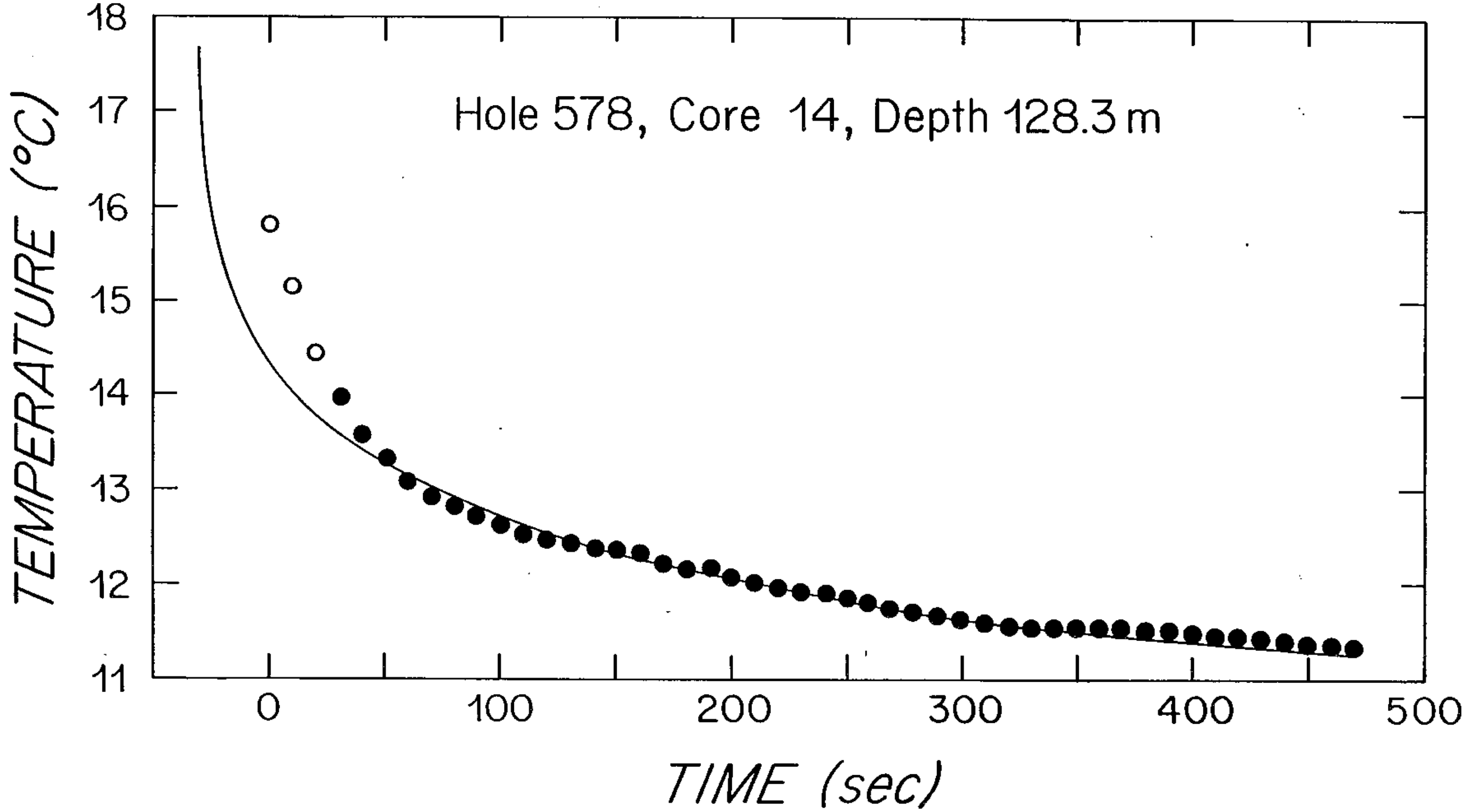

Figure 13. Thermal decay of HPC during core 14, hole 578, Leg 86 (see Fig. 9). Solid dots are data points fit on the theoretical curve shown.

extrapolated value. Probably the true in-situ temperature is much closer to the value extrapolated using the two-dimensional theory, as seems to be the case from the comparison of the curves (Fig. 13). This should be tested by attempting to leave the HPC in the bottom as long as possible so that the data may be compared with the theoretical extrapolation.

Another useful set of data is the temperature of the near-bottom water measured by the HPC during its passage through the drill pipe. The temperature of the water in the drill pipe is usually close to that of the surrounding sea water, so that a calibration of the instrumentation can be obtained for each measurement. This is particularly useful if different sets of instrumentation are used for measurements in the same hole. It provides a useful check on the accuracy of the data obtained, and also provides another temperature data point at the sea floor.

C. <u>TEMPERATURE CALIBRATION STABILITY</u>.

We have some limited data on the temperature stability of some units, based on accurate repeated calibrations. Altogether 10 hybrid electronic units were constructed. One was not calibrated, because it was left as a hybrid that could be adapted to fit the annular cavity of the new cutting shoe, with the possible addition of acoustic communication. Three others were lost during operations at sea shortly after they were constructed and calibrated; hence, no history of stability over time exists for these units. Of the six other units, four (units Nos. 5, 6, 7, and 8) were calibrated in the laboratory two or more times, which provides evidence of temperature stability.

Calibrations were performed with the entire unit immersed in a stable water bath, at 7 temperatures uniformily distributed at about 10°C intervals over the range 0 to 60°C, as described above. The differences between calibrations in 1983, over a period of about 11 months for units Nos. 5 and 6, and about 7 months for units Nos. 7 and 8, are shown in Fig. 14a. Similarly, the differences over about 11 months in 1984 for all 4 units are illustrated in Fig. 14b.

The equivalent temperature shifts shown by the 1984 calibrations are within .01 - .02°C over the entire temperature range for all units, and for 2 of the units (Nos. 7, 8) in the 1983 calibrations. These are probably within the resolution of measurement for these units. The larger shifts (.05 - .07°C) for units Nos. 5 and 6 in 1983 may correspond with their use on Leg 92 and subsequent legs. The shifts are fairly uniform over the entire temperature range, suggesting that the thermistor sensor (most likely) or a bridge resistor may have changed as a result of physical shocks associated with the measurements. (Note that units Nos. 5 and 6 shifted in the opposite sense from each other.) Unfortunately, the units used extensively on Leg 86 were lost in operations on subsequent legs before they could be recalibrated.

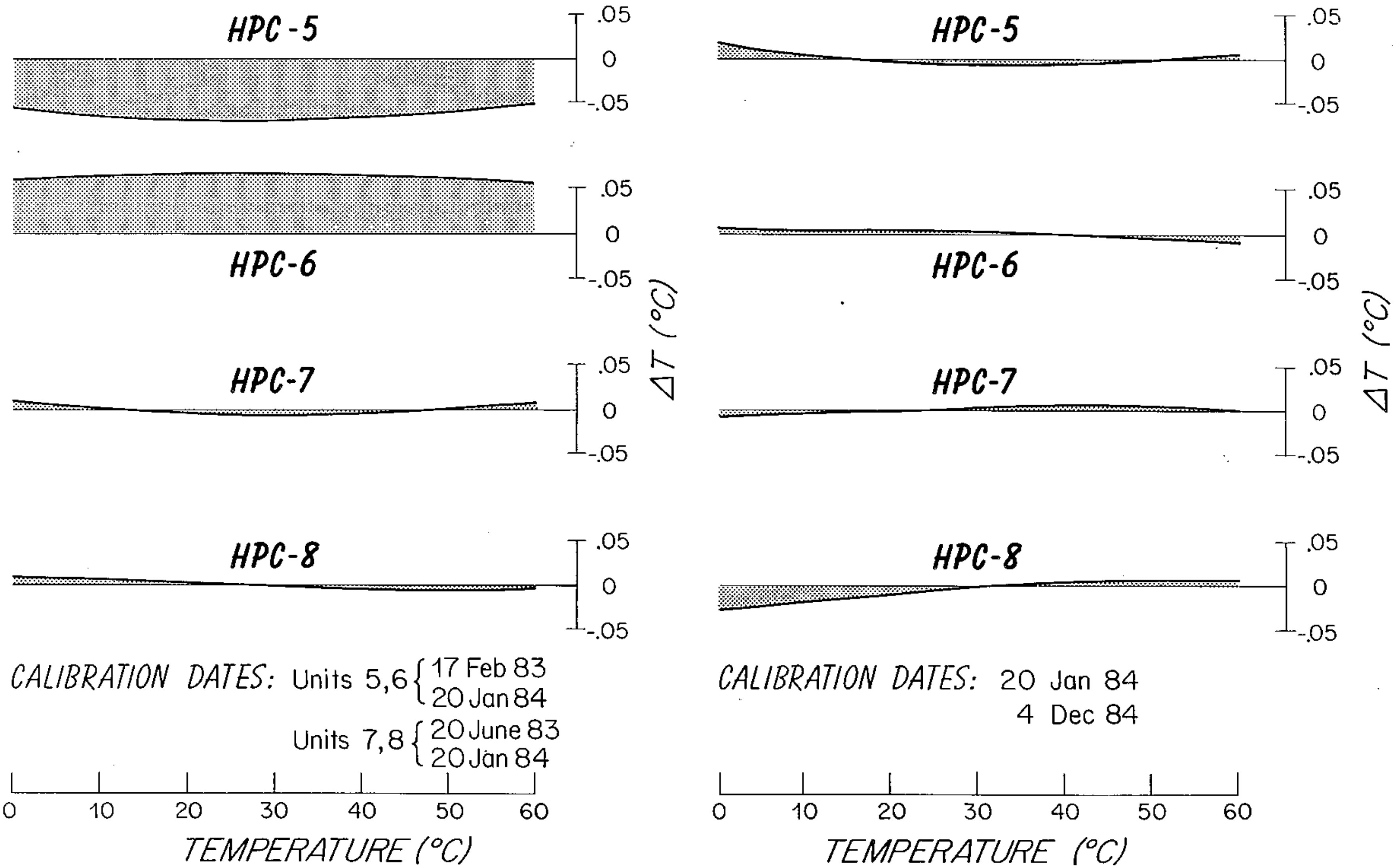

Figure 14. Comparison of temperature calibrations of 4 instruments before and after deployment and use at sea, over dates as indicated. Differences are calculated as the later calibration subtracted from the earlier.

VII. FUTURE

The current software for the temperature recorder provides a sampling interval in the range of 0.1 to 6553.5 sec. The 0.1 sec sampling interval corresponds to an upper frequency limit of 5 Hz (the Nyquist sampling rate) or 2 Hz (if you want to preserve the waveform). This rate would fill the memory in 130 seconds. The longest sampling interval of 6553.5 sec corresponds to a minimum period of 1.82 hr (at the Nyquist rate), or 4.55 hr (for waveform preservation). This slowest rate would take 2366 hr to fill the memory, which far exceeds the battery life (approximately 20 hrs.).

Other applications are possible for the temperature recorder. The input is an A/D converter, so it could be adapted to measure any voltage or resistance bridge measurement. A possible application is to measure the output of a strain gage type pressure transducer, or the voltage output of any sensor that has a voltage output. The A/D converter can sample up to 25 times per second, if the integrating capacitor is changed and the software modified. This implies an upper frequency limit of 12 Hz (the Nyquist rate) or 5 Hz (waveform preservation). The lowest frequency is limited by the memory size and the sampling rate. At 25 samples per second, the memory would be filled in 52 seconds. At the other extreme, data rates may be as long as desired to fill the memory (currently 1300 2-byte words), limited only by battery life.

The present coring shoe cavity has been enlarged in a new design, which has the same diameter as the original cavity, but it extends completely around the coring tool to form an annulus. It is 0.28 in. thick and 5 in. long. This larger cavity, if layed out flat, would be 10 in. long, by 5 in. wide, by 0.28 in. thick. It was possible because the strength of a shell (the cylinder) is much stronger than a flat plate. Actually, the small cavity is probably stronger that we calculated (see Appendix A), but we conservatively used the formula for a flat plate. This new cavity was made by welding the outer wall to the piece containing the cutting edge and the inner wall. It was no more expensive than the previous design, because no EDM (electro discharge machining) was necessary. It was designed at the DSDP at Scripps. There is some question about the strength of the cylinder wall under impact, but that can be tested. If it would fail under severe impact, you could take your chances, and accept a loss if it does fail. However you would not want it to be damaged, then deform under pressure and get stuck in the drill string.

The whole diameter of the coring tool can be used for instrumentation where only drilling and logging is being done and the material being drilled is too hard for HPC coring. This increases the space available for additional instrumentation.

Each time data is read from the temperature recorder, the coring shoe has to be opened up to enable connection to the instrument, and cleaned and closed up for the next use. This is troublesome, as well as risky by exposing the electronics to possible damage from salt water, or the chance of losing data by inadvertently disconnecting the battery.

The temperature recorder could use acoustic communication through the wall of the cavity in the coring shoe. A small transducer about the size of a dime could transmit using a 1 MHz carrier, using frequency shift keying. It could also receive instructions. With the large annular cavity, there would be plenty of room for batteries, certainly enough to last one leg (2 months), or possibly even for a year. So the instrument would not have to be opened up during use, or possibly not at sea. It could be sent back to the lab for new batteries.

At one time flexible printed circuits were considered as an alternative to the hybrid. A flexible circuit was tried in a rolled up circuit for another instrument, and it was discovered that the traces break where they are soldered to the integrated circuit packages. The flexible board industry laminates rigid boards to the flexible circuit in the regions where there are electronic parts, and uses the flexible part only to interconnect the rigid boards. If such a structure were used in the hollow annulus, it would take up most of the space. The resistance to physical shock is problematical. However, the hybrid is difficult and somewhat expensive to produce in small quantities, which might be alleviated with a more spacious layout using small outline package IC's on boards with flexible interconnections. Another possibility is combining multiple hybrids on separate hard boards, with flexible interconnections.

There has been a desire expressed to reach higher temperatures, perhaps 200°C. The present design has a nominal limitation of 75°C, determined by the temperature rating of the A/D converter. All other parts are rated to 125°C. The batteries can last a short time at 70°C. Eutectic solder (63% tin, 37% lead) melts at 183°C, and other ratios get mushy at this temperature.

To reach 200°C, the entire electronics circuitry and power supply would have to be redesigned with higher temperature parts. Such high temperatures would normally be encountered at depths or in regions (e.g., ridge crests) where the materials are too hard to use the HPC. If the entire diameter of the coring tool could be used for instrumentation, then existing higher temperature batteries (0.55 in. diameter) could be considered. Hybrid circuits at 200°C could probably be fitted into the 0.28 in thick space inside the wall of a coring tool. The thermometer would be a platinim resistance thermometer of some type of rugged construction. Absolute accuracy of 0.5°C seems possible at the higher temperatures.

VIII. ACKNOWLEDGEMENTS

Several persons have contributed significantly to this project. Ed Mellinger wrote the assembly language programs for the recorder and participated in the design of the electronics and construction of the breadboard. Skip Pelletier wrote the programs for the personal computers (Apple and Pro-350) for controlling the temperature recorder, and processing the recorder data. Dave Huey and Mike Storms (DSDP) designed and arranged for construction of the special coring shoe, Don Bellows (DSDP) did the same for the new annular cavity coring shoe. Charlie Yiakas and Dave Vacca (TSI) were always helpful and patient in consultation and for construction of the hybrid circuit. Ki-iti Horai gave the recorder its first extensive field tests on Leg 86 of the DSDP, and developed the theory to obtain equilibrium temperatures from transient thermal decay of the corer.

We are grateful for the financial support provided for various phases of this project:

1. USGS (Woods Hole) for the initial support to begin development of instrumentation (Project No. 17-18920.22); to the Ocean Industry Program, WHOI (Project No. 4471) to complete the development.

2. NSF, Grants OCE82-14658 and OCE83-00073 to enable construction of additional units for field measurements and for data analysis.

REFERENCES

Anonymous, Design and operation of an advanced hydraulic piston core, Tech. Rept. 21, Deep Sea Drilling Project, University of California, 269 pp., July 1984.

Anderson, R.N., M.A. Hobart, and M.G. Langseth, Geothermal convection through oceanic crust and sediments in the Indian Ocean, Science, 204, 828, 1979.

Anderson, R.N., M.G. Langseth, and J.G. Sclater, Mechanisms of heat transfer through the floor of the Indian Ocean, J. Geoph. Res., 82, 3391-3409, 1977.

Becker, K., and R.P. Von Herzen, Heat transfer through the sediments of the mounds hydrothermal area, Galapagos spreading center at 86°W, J. Geoph. Res., 88, 995-1008, 1983.

Erickson, A.J., and R.P. Von Herzen, Downhole temperature measurements and heat flow data in the Black Sea - DSDP Leg 42B, in: Initial Reports Deep Sea Drilling Project, vol. 42, part 2, by D.A. Ross, Y.P. Neprochnov, et al. (U.S. Govt. Printing Office), 1085-1103, 1978.

Erickson, A.J., R.P. Von Herzen, J.G. Sclater, R.W. Girdler, B.V. Marshall and R. Hyndman, Geothermal measurements in deep-sea drill holes, J. Geoph., Res., 80, 2515, 1975.

Horai, K. and R.P. Von Herzen, Measurement of heat flow on Leg 86 of DSDP, in Initial Reports of DSDP, vol. 86, Washington (U.S. Government Printing Office), 759-777, 1985.

Roark, R.J. and W.C. Young, Formulas for Stress and Strain, Fifth edition, McGraw-Hill, New York, 1975.

Steinhart, J.S., and S.R. Hart, Calibration curves for thermistors, Deep Sea Research, 15, 497-503, 1968.

Von Herzen, R.P., Geothermal measurements, Leg 21, in: Initial Reports Deep Sea Drilling Project vol. 21, by R.E. Burns, J.E. Andrews, et al. (U.S. Govt. Printing Office), 443-457, 1973.

Von Herzen, R.P., J. Crowe, and K.E. Green, The eastern equatorial Pacific ocean heat flow low: evidence of hydrothermal circulation, ms. in preparation.

Yokota, T., H. Kinoshita, and S. Uyeda, New DSDP downhole temperature probe utilizing IC RAM elements, Bull. Earthquake Res. Inst., 55, 75, 1980.

APPENDIX A. CAVITY STRENGTH CALCULATIONS

1. Kidney shaped cavity (currently used).

The cavity had to fit into the 9/16 in. thick wall of the coring shoe, and operate at 21,000 ft. deep (10 kpsi) with some margin of safety. A very strong heat treatable stainless steel, Nimark 250, from Carpenter Technology, Carpenter Steel Division, Reading, PA, with a typical yield strength of 250 kpsi, was used for the portion of the coring shoe containing the cavity.

We selected a hybrid package to fit in the cavity. We made the width just large enough to accomodate the hybrid package (0.80 in.) and then made the cavity as high as possible. The actual cavity walls are curved (Fig. 6), so although the height of the cavity is 0.28 in., the height of a rectangular hybrid package inscribed in the curved cavity is less. The length of the cavity is 5 in. to accomodate the combined package of hybrid and printed circuit board. A cylindrical hole 0.125 in. in diameter extends beyond the end of the cavity to accomodate the thermistor probe.

A flat plate approximation was used for the curved walls of the cavity. This approximation is conservative, in that the curved shape should be stronger. The width of the cavity used for the flat plate approximation is the portion of the arc of the outer wall that has uniform wall thickness, measured along the outside surface of the coring tool. The equation below is derived from formulas for flat plates with straight boundaries and constant thickness (Roark and Young, 1975, p. 386, Table 26).

The rectangular plate, shown in Figure A-1, has three edges fixed and one edge simply supported. The force is uniform over the entire plate. Using Table 26, Case 9 (Roark and Young, 1975, p. 394), the general form for the stress is:

$$s = \frac{B\ q\ b^2}{t^2} \qquad \text{(A.1)}$$

where a, b = the dimensions of the plate; t = thickness of the plate; q = unit force on the plate; and B is a variable which depends on the ratio a/b and the direction and location of the stress. B is taken from a table in the above reference which had to be extended to a/b = 0.16 (vs. a minimum of 0125 in the table) by graphical extrapolation on log-log graph paper.

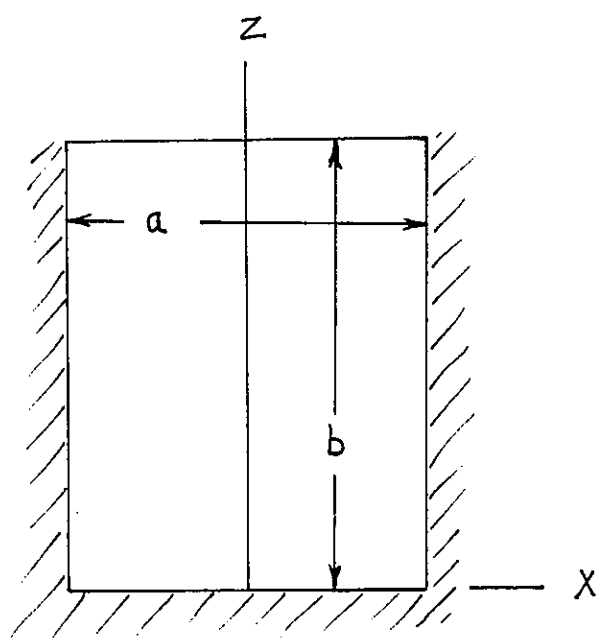

Figure A-1. Flat plate model used for cavity strength calculations.

For $q = 10{,}000$ psi, $a = 0.800$, $b = 5.00$, $t = 0.145$, the maximum stress in the a direction (circumferentially) occurs at $x = \pm a/2$, $z = 0.6\,b$, for which $B = 0.0133$,

$$s_a = \frac{(0.0133)(10^4)(5)^2}{(0.145)^2} = 158{,}200 \text{ psi}$$

The maximum stress in the b direction (axially) occurs at $x = 0$, $z = 0$, for which $B = 0.0083$,

$$s_b = \frac{(0.0083)(10^4)(5)^2}{(0.145)^2} = 98{,}690 \text{ psi}$$

The largest stress is 158 kpsi in the outer wall, which is about 70% of the minimum yield strength (230 kpsi) of the material. The curvature of the plate serves to strengthen the plate further, but is not included in the calculation. The coring shoe was tested at and survived 10 kpsi, the working pressure. It was not tested at a higher pressure for fear of collapsing the cavity. However, we infer from the calculations for the strength of an annular cavity (see following) that the actual strength of this cavity design may be considerably stronger.

The cavity for the battery was made the same size as the electronics, except for length. It is 3 in. long, while the cavity for the electronics is 5 in. long.

An annular cavity, the same thickness as the electronics cavity and 0.25 in. deep, provides space for compression of the air when the two parts of the coring shoe are screwed together. Without this space, the air compressed in the cavity would bend the top of the lid of the hybrid package and short out the hybrid circuitry.

2. Annular Cavity.

A much larger space for the electronics can be achieved if an annulus were hollowed out of the interior of the coring shoe. A cylinder is stronger than a flat plate. A short cylinder is stronger than a long cylinder because the support from the ends has more influence on a short cylinder. The cylinder can fail either in collapse or in buckling. Different equations are used for each case, and the lowest pressure at which the cylinder fails is the one that is used. A number of lobes are formed by the tube in buckling. The buckling pressure for each lobe number is calculated and the lowest buckling pressure is the one used.

The dimensions of the original coring shoe (Fig. 6) were used, which would make the same thickness available for the electronics. First the collapse equations are used, which result in a collapse pressure of 25,350 psi for the inner cylinder and 17,925 psi for the outer cylinder. Then the pressure for buckling the cylinder for a given number of lobes is calculated versus the cylinder length. Plots were made of pressure vs. length for each number of lobes. Then, using the minimum buckling pressure of all the curves, length or pressure can be chosen. We chose 5 in. long, which gave a buckling pressure of 18,000 psi, with 3 lobes.

Cylinder collapse strength.

The walls of the annulus are concentric cylinders, each considered as a short supported tube (Fig. A-2). The cylinders are fixed at one end and simply supported at the other.

The material used is Nimark 250, with the following characteristics:

Min. yield strength = 230 k pounds/in^2
Modulus of elasticity, E = 27.5 x 10^6
Poissons ratio, v = 0.3

The equations below are from Table 32 (Roark and Young, p. 504): Formulas for a thick walled vessel under internal and external loading. Notation: a = outer radius; b = the inner radius; s_2 = normal stress in the circumferential direction; v = Poissons ratio.

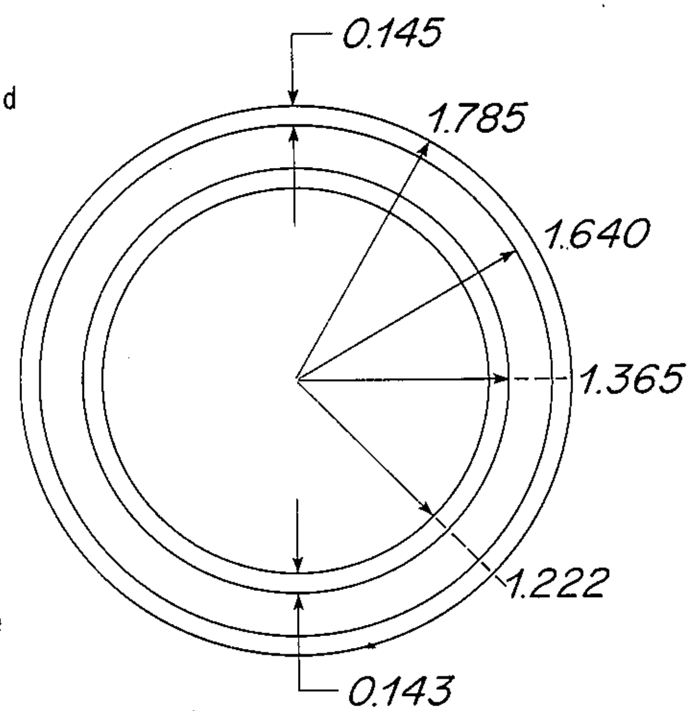

Figure A-2. Cross section view of the hollow annulus.

The inner cylinder has uniform internal pressure, q lb/in^2, in all directions; ends capped. Using Table 32, Case 1b (Roark and Young, p 504),

$$\max s_2 = q \frac{a^2 + b^2}{a^2 - b^2} \quad \text{at} \quad r = b \qquad (A.2)$$

For a = 1.365
b = 1.222

$$\max s_2 = (10^4) \frac{(1.365)^2 + (1.222)^2}{(1.365)^2 - (1.222)^2} = 25{,}350 \text{ psi}$$

The outer cylinder has uniform external pressure, q lb/in^2, in all directions; ends capped. Using Table 32, Case 1d (Roark and Young, p. 504),

$$\max s_2 = \frac{2\,q\,a^2}{a^2 + b^2} \quad \text{at} \quad r = b \qquad (A.3)$$

For $a = 1.785$
$b = 1.640$

$$\max s_2 = \frac{2\,(10^4)\,(1.785)^2}{(1.785)^2 - (1.640)^2} = 17{,}925 \text{ psi}$$

Elastic instability calculations.

The outer cylinder is subject to elastic instability because it has external pressure. A cylinder with supported ends has a higher pressure for elastic instability, as it becomes shorter.

The equation below is from Table 35, Formulas for elastic stability of plates and shells (Roark and Young, p. 556). Notation: E = modulus of elasticity; v = Poisson's ratio; and t = thickness for plates and shells.

The thin tube has closed ends under uniform external pressure, lateral and longitudinal (length of tube = ℓ; radius of tube = r). The ends are held circular. Using Table 35, Case 20 (Roark and Young, p. 556), where q' = external pressure at which elastic buckling occurs:

$$q' = \frac{E\left(\frac{t}{r}\right)}{1 + \frac{1}{2}\left(\frac{\pi r}{n\ell}\right)^2} \left\{ \frac{1}{n^2\left[1 + \left(\frac{n\ell}{\pi r}\right)^2\right]^2} + \frac{n^2\,t^2}{12\,r^2(1-v^2)}\left[1 + \left(\frac{\pi r}{n\ell}\right)^2\right]^2 \right\}$$

here n = number of lobes formed by the tube in buckling. As described by Roark and Young, to determine q' for tubes of a given t/r, a group of curves is plotted for each integral value of n of 2 or more, with ℓ/r as ordinate and q' as abscissa; that curve of the group which gives the least value of q' is then used to find the q' corresponding to a given ℓ/r.

Plots of this equation are shown in Fig. A-3, where pressure is plotted vs. length of the cylinder, for r = 1.785 in. and t = 0.145 in.

The buckling pressure is 18 kpsi, for a length of 5 in., which is nearly the same as the collapse pressure for this shape. Thus, the walls of the 5 in. long cylindrical annulus are more than strong enough to withstand 12 kpsi at the bottom of the drill string.

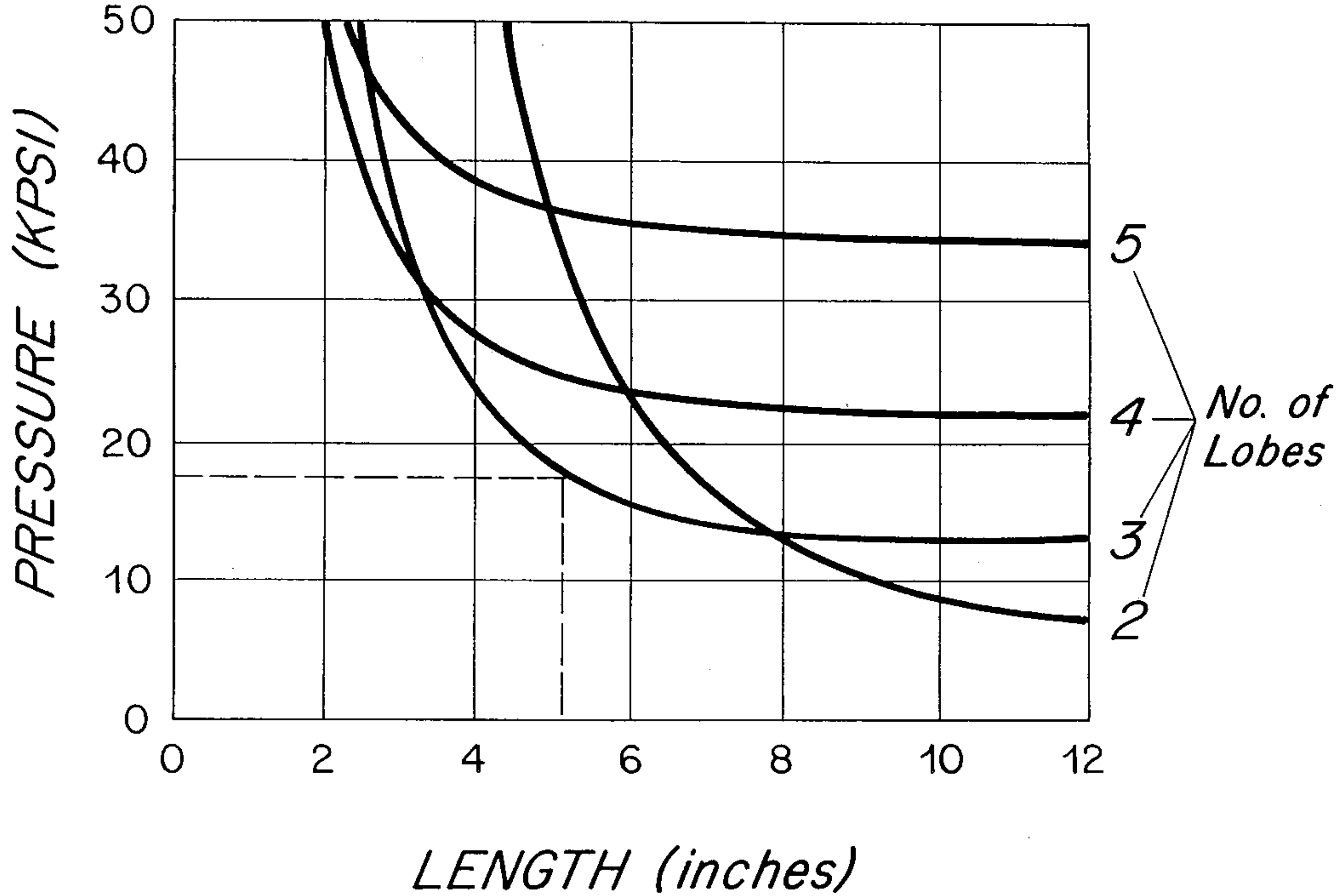

Figure A-3. Buckling pressure vs. length of cylinder, for various numbers of buckling lobes.

APPENDIX B. DETAILED DESCRIPTION OF ELECTRONICS

1. Description of Block Diagram.

The block diagram (Fig. 4) corresponds very closely in structure to the actual schematic. The temperature recorder consists of 7 blocks plus battery: the microprocessor, RAM, UART, A/D converter, thermistor bridge, power supply, and battery. The measurement begins with the thermistor bridge, with a voltage nearly linear with temperature. The A/D converter digitizes this voltage with 13 bit resolution. The microprocessor instructs the A/D converter when to convert and stores the measurement in memory. The UART is used to send the stored data serially to an awaiting external device (a computer). The UART is also used to load the program into RAM for 2-way communication with the computer.

A voltage regulator provides 5 volts to the electronics from an 8 volts battery. A 4 volts mode is available to reduce power consumption. The power switch controls the 5 volts to the A/D converter, so that it may be shut down when not being used, in order to save power. The voltage inverter converts some of the 5 volts to -5 volts for the A/D converter.

2. Detailed Description of the Schematic (Fig. B-1).

The 1802 microprocessor controls all the rest of the elements of the temperature recorder. A crystal of 38.4 kHz runs with the internal oscillator of the microprocessor to provide the computer clock. A low frequency was chosen to save power, since the microprocessor does not have to work very fast. The clock frequency was chosen so that a binary divider chain can provide the UART clock input, which is 16x the UART baud rate.

There are 3 kbytes of RAM, controlled thru the 1866 memory selector, by the address lines of the microprocessor. The 8 address lines are multiplexed. The high order byte comes out first and the low 4 bits are stored in the 1866. Two bits drive the RAMs directly and two are decoded to select the RAMs. The low order byte comes out second and drives the address lines of the RAMs directly. The RAMs are 1 k x 4 bits, so 2 RAMs are used simultaneously: one for the upper four bits of data, and one for the lower 4 bits of data. A write enable line from the computer is active when writing to memory. The address lines and the inhibit for the 1866 selector come outside the hybrid so it can be tested, more memory added or other memory substituted for the internal memory.

The UART runs at 1200 baud, and provides 2-way communication thru the 9 pin recorder connector. The clock for the UART is 19.2 kHz, which is 16x the 1200 baud rate, derived from a binary divider on the computer clock. The baud rate can be changed on the printed circuit board.

The UART 8 bit data busses for transmitting and receiving are connected to the microprocessor data bus. The transmit buffer is loaded by I/O line 5, and the receiver buffer is read by I/O line 4. These lines come from the 1853 N line decoder. When data has been transmitted, the transmit buffer

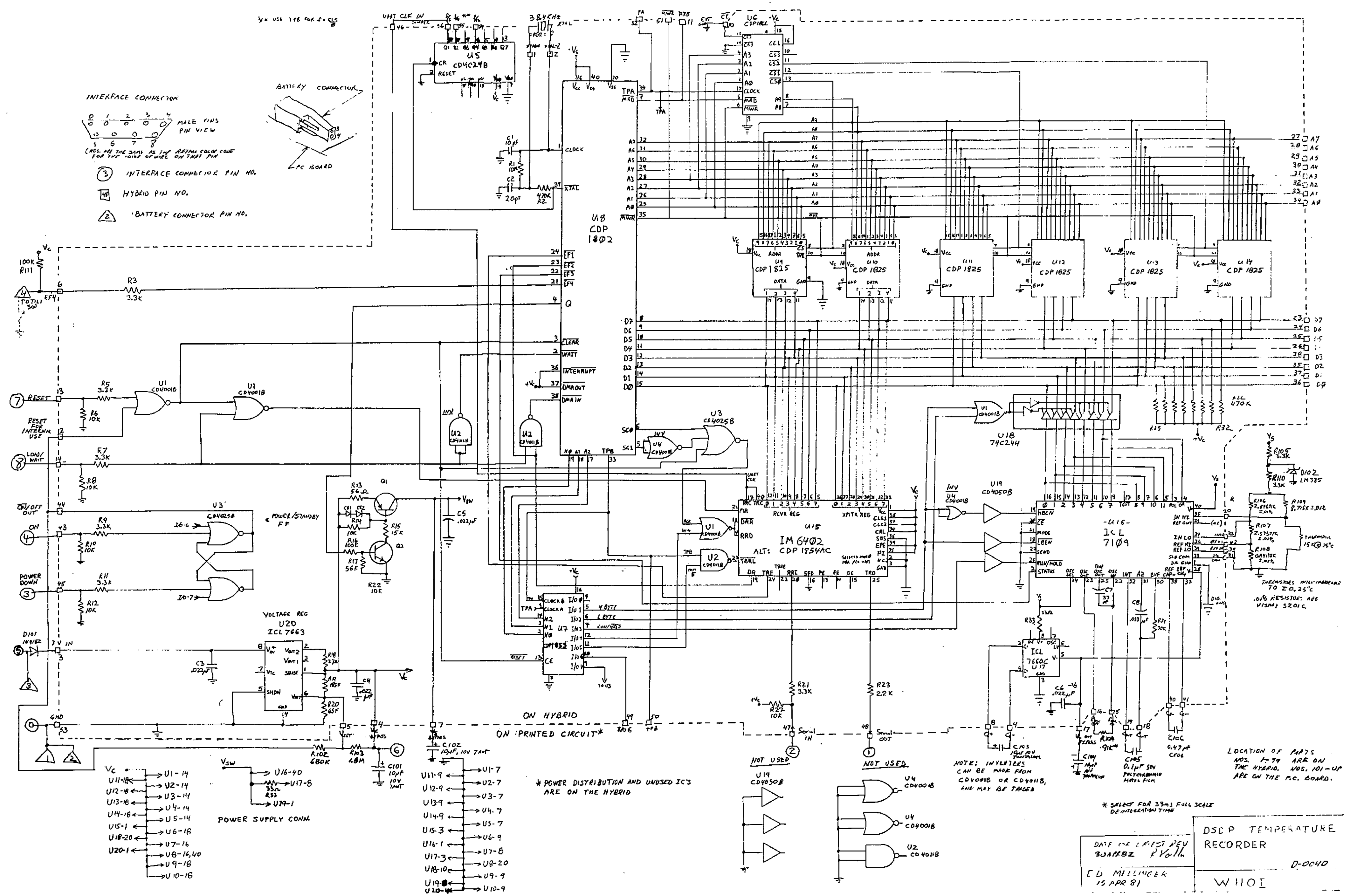

Figure B-1. Schematic of the temperature recorder.

empty line raises (to inverted logic 0) the external flag 1 input (EF1-bar) of the microprocessor. When the receive buffer is full, the data ready line raises the external flag 2 input (EF2-bar) to the microprocessor. The microprocessor has to test these flags to determine the state of the UART.

A dual slope type A/D converter is used to measure the bridge output. Its integration period is set to 1/60 second to average out 60 Hz pick-up while making lab measurements. The A/D converter averages out noise over the integration period, which means it is less sensitive to noise.

The run/hold line for the A/D is controlled by I/O line 3. This line, as well as the high byte enable and low byte enable for data output are driven by buffers that get their power from the A/D power, but do not load the driving source when the A/D power is switched off.

The reading of the high byte out of the A/D is enabled by I/O line 1, and the low byte by I/O line 2. A tri-state buffer is used between the A/D data output and the data bus so the A/D converter can be turned off without pulling the data bus down.

The bridge circuit for the thermistor is made of Vishay (Malvern, PA) resistors, which are very stable and temperature insensitive (<1ppm/°C). The bridge runs at a low voltage so that only about 10 m°C self heating of the thermistor occurs. The self heating, which varies about 30% over the temperature range, is included in the temperature calibration. A zener-like voltage regulator provides 2.5 volts to the bridge to keep the small fluctuations (25 mV) in the 5 volts line from changing the voltage to the bridge during conversion.

The inactive leg of the bridge also provides the reference voltage for the A/D converter, which is set to about 200 mV. A ratiometric measurement is made from the bridge voltage, so the exact voltages are not important. A convenient feature of this A/D converter is the Input Lo terminal, a floating high impedance input, which allows us to center its operating range in the middle of the bridge output swing. Thus we can use both the positive and negative range of the 12 bit plus sign A/D converter to obtain a 13 bit conversion.

An I/O instruction has numbers 1 thru 7, which is encoded into 3 bits on the N0, N1, and N2 lines. The 1853 N line decoder decodes these 3 bits into 8 lines for operating various I/O devices. These are:

1. IN 1 - high byte enable on A/D.
2. IN 2 - low byte enable on A/D.
3. OUT 3 - Run/Hold on A/D.
4. IN 4 - read UART receive buffer.
5. OUT 5 - load UART transmit buffer.
6. OUT 6 - raise power supply voltage to 5 volts (also available as an output pin for control of an external device).
7. OUT 7 - lower power supply voltage to 4 volts.

Although these lines can be programmed to be inputs or outputs, they are wired to do only one or the other (except for the potential use of I/O line 6).

A very low power voltage regulator supplies a regulated 5 volts to the electronics, when the computer is running normally. It can be switched by I/O line 7 to regulate at 4 volts for standby to hold the data at about 1/2 the current (about 0.5 mamp) as at 5 volts. Under program control the voltage is switched to 4 volts after the data has been acquired. The computer seems to run at 4 volts, but proper operation of the computer and memory is not guaranteed below 4.5 volts. The power can be raised to 5 volts with I/O line 6.

To further save power after the data has been acquired, an Idle instruction stops the microprocessor, while the clock continues to run. This chops the current by another factor of 2 to about 0.25 mamp when the supply voltage is 4 volts. As it is wired up (no interrupts), the microprocessor cannot get out of the idle condition by itself, but must be reset externally.

A power switch, controlled by the Q line of the microprocessor, switches the +5 volts for the A/D converter. A 7660 voltage inverter, powered by this switched +5 volts, generates the -5 volts for the A/D converter. The A/D converter consumes about 2.5 mamp when on, so considerable power is saved when it is turned off.

The 1802 has a convenient Load mode, using the DMA IN input. Data from the UART is read into the memory, starting at address 0000_H, when in load mode and the DMA IN line is low. The Data Ready line of the UART controls the DMA IN line (via a gate) when in the Load mode. The load mode is entered by raising the reset and the load line in that order.

APPENDIX C. DETAILED LIST OF INPUT/OUTPUT INSTRUCTIONS

Details on the Temperature Recorder Computer Instructions.

Below are instructions specific to this particular computer. See the RCA 1802 programming manual for the general programming of the 1802.

The memory is 3 kbytes (3 x 1024), which is addressed sequentially from 0000_H through $0BFF_H$ (3071_D). The same memory is addressed each 4 kbytes up to 64 kbytes, because the address decoder does not look at the upper 4 bits.

Input/Output Instructions

IN 1	read A/D high byte
IN 2	read A/D low byte
OUT 3	run/hold-bar A/D converter (it takes 1/15 sec to make a conversion)
IN 4	read UART receive buffer
OUT 5	load UART transmit buffer
OUT 6	raise power supply voltage to 5 volts
OUT 7	lower power supply voltage to 4 volts

Other Instructions

SET Q	switch power on to A/D
RESET Q	switch power off to A/D

FLAGS (note: actual flag inputs are inverted logic)

EF1	UART transmit buffer empty
EF2	UART receive data buffer full
EF3	A/D status: 1 = running; 0 = finished, or ready to start
EF4	tilt switch vertical (start program)

Load Memory with Program

Raise reset line to 5 volts, then raise load line to 5 volts.

Now any input to the UART will go into memory. All 8 bits from the UART input go into memory.

To terminate load, lower the load line to 0 volts, then lower the reset line to 0 volts.

Now the program will start at location 0000.

UART

The UART runs at 1200 baud, 8 bits, no parity. 1 = 5 volts, 0 = 0 volts.

APPENDIX D. BRIDGE CIRCUIT EQUATIONS

The bridge circuit for the thermistor is shown in Fig. D-1.

E_1 is the input voltage to the bridge.

E_{IN} is the voltage into the A/D converter. The negative input to the A/D converter is derived from the inactive leg of the bridge, and is at the center of the bridge excursion.

E_{REF} is the reference voltage into the A/D converter. The reference voltage is 1/2 the full scale input to the A/D converter. This reference voltage is derived from the inactive leg of the bridge, instead of from a separate divider.

The A/D converter is ratiometric, that is, measures E_{IN}/E_{REF}. Therefore, the value of E_1 is not important, as long as it is stable shortly before and during a measurement.

The circuit can easily be solved for R_2, the thermistor resistance, as follows:

$$R_2 = \frac{R_1\left(\frac{R_4}{R_5} + 1 + \frac{E_{in}}{E_{ref}}\right)}{\left(\frac{R_3}{R_5} - \frac{E_{in}}{E_{ref}}\right)} \qquad (D.1)$$

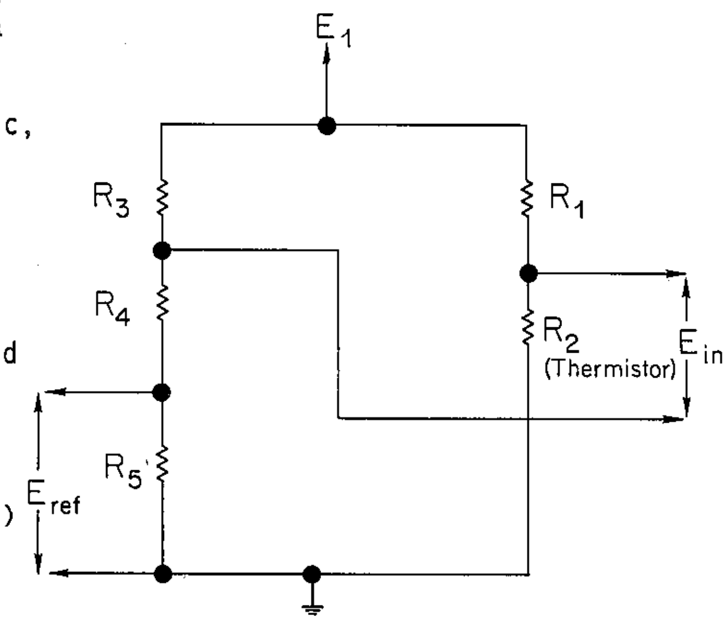

Figure D-1. Thermistor bridge circuit.

or

$$R_2 = 8.715 \left[\frac{\left(3.734039 + \frac{count}{2048}\right)}{\left(3.023975 - \frac{count}{2048}\right)}\right]$$

with the following circuit values (Fig. D-1):

R_1 = 8.715 K
R_2 = thermistor resistance
R_3 = 2.84671 K
R_4 = 2.57377 K
R_5 = 0.94135 K

$$\frac{E_{in}}{E_{ref}} = \frac{\text{count from A/D}}{2048}$$

APPENDIX E. CURVE FITTING TO THERMISTOR CALIBRATIONS

We have found that the thermistor calibrations are fit very well (to better than a few millidegrees over a 60° range) by a 3-parameter equation (Steinhart and Hart, 1968)

$$T^{-1} = A + B \ln R + C(\ln R)^3$$

where T = temperature (K)
R = thermistor resistance (ohms)

A, B, C are constants to be determined for each thermistor. Obviously, at least 3 calibration values are needed to determine the 3 constants in the equation, but usually we have an overdetermined problem of finding the best solution for more than 3 values. This solution is found by the usual least-squares method. Notice that the squared term of the power series is absent, so the usual curve fitting routines for a power series are not used. The least-squares method for this equation, assuming equal weighting of calibration values, is derived as follows:

We simplify the notation by solving for the equation

$$x = A + By + Cy^3$$

where we have substituted x for T^{-1}, and y for ln R. Then we want to determine the constants A, B, and C such that $\sum^{n} (A + By + Cy^3 - x)^2 =$ minimum for n calibration values. This condition is met by differentiating this expression with respect to each of the constants and setting equal to zero:

$$\frac{\partial}{\partial A} = 2 \sum (A + By + Cy^3 - x) = 0$$

$$\frac{\partial}{\partial B} = 2 \sum (A + By + Cy^3 - x)\, y = 0$$

$$\frac{\partial}{\partial C} = 2 \sum (A + By + Cy^3 - x)\, y^3 = 0$$

where the summation over n is assumed

or:

$$\sum (A + By + Cy^3 - x) = 0$$

$$\sum (Ay + By^2 + Cy^4 - yx) = 0$$

$$\sum (Ay^3 + By^4 + Cy^6 - y^3x) = 0$$

Re-arranging terms:

$$An + B\sum y + C\sum y^3 - \sum x = 0$$

$$A\sum y + B\sum y^2 + C\sum y^4 - \sum yx = 0$$

$$A\sum y^3 + B\sum y^4 + C\sum y^6 - \sum y^3 x = 0$$

Rewriting in matrix format:

$$\begin{bmatrix} n & \sum y & \sum y^3 \\ \sum y & \sum y^2 & \sum y^4 \\ \sum y^3 & \sum y^4 & \sum y^6 \end{bmatrix} \cdot \begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} \sum x \\ \sum yx \\ \sum y^3 x \end{bmatrix}$$

or:

$$\left[D_{ij}\right] \cdot \left[\beta_i\right] = \left[E_i\right]$$

The desired solution is therefore:

$$\left[\beta_i\right] = \left[D_{ij}\right]^{-1} \cdot \left[E_i\right]$$

APPENDIX F. PROGRAM LISTINGS FOR THE TEMPERATURE RECORDER

1. INSTRUCTIONS FOR TREC

NAME: TREC/AVRG

PURPOSE: To control operation of DSDP Coring Tool Temperature Recorder

MACHINE: 1802 COSMAC

SOURCE LANGUAGE: RCA CSDP Assembler

DESCRIPTION:

The Coring Tool Temperature Recorder is a miniature data aquisition system packaged in a custom hybrid integrated circuit. It was developed at WHOI for the purpose of measuring in situ temperature of deep ocean sediments, from the cutting shoe (tip) of a Hydraulic Piston Coring (HPC) tool. The recorder contains an 1802 microprocessor, 3k bytes of static RAM, a serial I/O port, and an A/D converter on a substrate approximately 0.55" x 2.7". An external sensor (nominally a thermistor bridge) and battery pack complete the recorder, which has the capability to aquire 1200 to 1500 fourteen bit data points at intervals of 0.2 seconds to several hours.

TREC is an operating program designed to drive this recorder. During initialization it allows the user to interactively specify one to eight sets of data samples to be aquired by the recorder, with independent turn on delay, sample interval, and sample count for each sample set. After initialization the sample parameters are echoed back to the user for verification, and the recorder's program memory is checksummed. Upon user command TREC enters aquisition mode, and each sample set is aquired using the specified delay, sample count, and sample interval parameters. TREC terminates aquisition and places the recorder in a low power idle mode after all specified sample sets have been aquired, or when the available data memory becomes full. After recovery of the recorder, the recorder CPU is reset and TREC outputs the stored data values to an external data storage device.

OPERATION:

LOADING

1. Place the recorder in Load mode and load the TREC binary code file into the recorder from the support computer. Detailed instructions for this proceedure will be provided for each type of support computer which is used.

2. Reset the recorder CPU to begin execution of TREC. Again, instructions will vary depending upon the support computer provided.

INITIALIZATION

3. TREC will prompt the user for three parameters for each sample set: the number of samples, the sample interval, and the delay period prior to sampling. Each parameter is input as a 0-4 digit hex number followed by a carriage return, as further specified in the "Input" section below. These sample parameters are stored in the recorder memory and later used to control operation of the recorder during data aquisition. The sample parameter prompts are:

SAMPLES ?   Input the desired hex number of samples for this sample set. The sum of samples for all sample sets should be less than 1380 decimal (564 hex). Entering a value of zero samples terminates the initialization proceedure and causes verification to begin.

INTERVAL ?   Input the desired hex sample interval for this sample set, in tenths of seconds, from a minimum of 2 up to a maximum of 65535 decimal (2-FFFF hex). Note: a specified interval of 1 (0.1 sec) will result in an actual value of 65536 (6553.6 sec).

DELAY ?   Input the desired hex value of turn on delay prior to this sample set, in seconds, from 0 to 65536. Turn on delay is the delay before aquiring the samples specified by # SAMPLES and INTERVAL.

VERIFICATION:

When a value of zero samples is input during initialization, input stops and verification begins. Each of the sets of sample parameters is read from memory and output for inspection by the user. A simple checksum is computed over program memory and output also. The user is then prompted for further action. A typical verification output would be:

```
#SAMPLES = 0080   INTERVAL = 0064   DELAY = 1C02
#SAMPLES = 0400   INTERVAL = 000A   DELAY = 0000
#SAMPLES = 00E4   INTERVAL = 0032   DELAY = 0000

CHECKSUM = 0058   ACTION ?
```

(This would specify an inital delay of 120 minutes followed by 128 samples at 10 second intervals, 1024 samples at 1 second intervals, and 228 samples at 5 second intervals, followed by power down.)

Examine each displayed parameter to be sure it is correct, and compare the checksum to the correct value for the TREC version in use. If all information is correct, typing a capital "G" in response to the "ACTION ?" prompt will cause TREC to enter aquisition mode and begin the first delay period. Typing "G" may be delayed for as long as necessary, to synchronize the pending aquisition cycle with external events such as HPC operation.

If any of the displayed sample parameters are incorrect or must be changed, TREC may be re-initialized by typing a carriage return in response to the "ACTION" prompt, and returning to step 3 above.

If the checksum does not match the correct value given for the TREC version in use, a data error has occurred during or after program load. Do not use a program with a checksum error. Reload TREC from the support computer and start over.

AQUISITION:

During the delay period prior to sampling, TREC outputs an "@" symbol every second to verify correct program operation. Once the "@" is observed the instrument may be sealed and deployed using appropriate proceedures. The tilt switch is active during the delay periods, as follows: If the recorder is horizontal continuously for more than 1 minute, the delay timing stops ("@" stops also). Delay timing restarts from the initially specified delay value for the current sample set, after the recorder has been restored to vertical continuously for 5 minutes.

After the specified delay period has elapsed the A/D converter is powered up and the number of samples specified for the current sample set is aquired, at the interval specified. After sampling, the number of samples in the next sample set is read from memory. If it is zero, the recorder is powered down and the CPU executes an "Idle" instruction until instrument recovery. If the number is not zero, the remaining sample parameters for the next sample set are read from memory and the next delay period begins. The A/D converter is powered down during delay unless the delay is zero seconds. If data memory becomes full during sampling further sampling is inhibited and the recorder is powered down as usual.

PLAYBACK:

Upon recovery the recorder will normally have exited Aquisition mode and be powered down with the CPU reset. After power up and release from reset TREC will dump the stored data as a sequence of 4 hex digit records, one for each sample aquired. When all samples have been dumped the message "STOP" is output and the recorder is powered down. The data playback sequence may be restarted after power down by again resetting the recorder, using the playback routine on the support computer.

INPUT FORMATS:

All input is serial asynchronous ASCII at 1200 baud; 8 data bits, one stop bit, no parity bit.

SAMPLE PARAMETERS. Each parameter is input as a zero to four digit hex number followed by a carriage return. Valid hex digits are echoed back to the user input device; invalid or unrecognized characters are not. The carriage return is not echoed. Valid digits are 0-9 and A-F. If no digits are input before the carriage return a default value of zero is assumed for the parameter. Only the last 4 digits entered are significant, so that typing errors may be corrected by typing unitl the last 4 digits represent the desired value and then typing carriage return.

OUTPUT FORMAT:

All output is serial asynchronous ASCII at 1200 baud; 8 data bits, one stop bit, no parity bit.

PROMPTS. The prompt "#SAMPLES?" is output upon initial reset, followed by the "INTERVAL?" and "DELAY?" prompts. The set of three prompts is repeated for each sample set specified. The prompt formats are:

(CR)(LF)#SAMPLES?(SP)(ETX)
(SP)(SP)(SP)INTERVAL?(SP)(ETX)
(SP)(SP)(SP)DELAY?(SP)(ETX)

where LF = line feed (hex 0A), CR = carriage return (hex 0D), SP = space (hex 20), and ETX = End-Of-Text (hex 03). A 200 millisecond pause follows each CR transmitted by TREC.

VERIFICATION. A double linefeed is output on entry to the verification routine. A verification record is output for each set of sample parameters entered during initialization. Each record has the format

#SAMPLES= HHHH INTERVAL= HHHH DELAY= HHHH(CR)(LF)

where HHHH is a four digit hex number. After the last record the checksum is output in the format

CHECKSUM= HHHH ACTION?(SP)(ETX).

AQUISITION. During the delay intervals the character "@" is output once per second. This output stops during tilt switch waits and during the actual sampling process.

PLAYBACK. After recovery and CPU reset the recorded data is output. Each 14 bit sample is output as four hex digits H_i in the following format

$H_3H_2H_1H_0$(LF)(CR)

where: H_3 contains the overrange (bit 2) and polarity (bit 3) bits; bits 0 and 1 of this digit should be ignored.
H_2 contains bits 11 (MSB) through 8 of the A/D output,
H_1 contains bits 7 through 4,
H_0 contains bits 3 through 0 (LSB).

Each carriage return is followed by a 200 millisecond pause. After the last aquired sample has been output, the end of data message is output:

STOP(LF)(CR).

ERROR MESSAGES:

TREC performs minimal input validity checking due to its small size. If a non hex character is supplied during data input, that character will not be echoed to the user. No error message is output. Input values are not checked for reasonableness or consistency.

RESTRICTIONS:

1. The sum of the number of samples in all sample sets should not exceed 1380 decimal. No harm will result from specifying a greater number of samples, but only the first 1380 samples will be aquired.

2. Specifying a sample interval of "1" (0.1 sec) will result in an actual sample interval of 6553.6 seconds.

SUBROUTINES REQUIRED: (all internal to TSEC package)

The first set of routines listed below are memory resident before, during, and after aquisition.

PLAYBK	-	dumps recorded samples.
OUTDAT	-	outputs a 4 hex digit number.
MAIN	-	aquisition control program.
DLAY	-	turn on delay; tilt switch.
AQUI	-	sample and sample interval control.
GETDAT	-	gets one sample from A/D.
TSEC	-	precision 0.1 second timing routine.
OUTCHR	-	outputs a variable length ASCII string.

The following routines are loaded into the data area of memory and are executed once during initialization. They are overlaid (destroyed) by data during the actual aquisition process.

INIT	-	initializes registers, prompts for parameters.
VERIFY	-	outputs parameters, computes checksum.
INDAT	-	reads and converts 4 digit ASCII hex numbers.

DATA STRUCTURES:

Most operating parameters, counters, and pointers are stored in the internal CPU registers. There are three simple data structures:

1. STACK. Used for temporary storage during computations and for holding data to be output with the OUT instruction.

2. PARAMETER STRINGS. There may be one to eight six-byte strings. Each string consists of three 16 bit (two byte) values stored high byte first: number of samples, interval, and delay. A value of zere samples indicates the end of the string area.

3. DATA VALUES. Two-byte data values are stored high byte first, beginning directly above the last parameter string and continuing up through the highest location in the available memory.

TIMING:

All TREC timing routines are based on a 38.400 kHz CPU clock. Assuming this clock frequency, DLAY and AQUI will execute their respective delay and sample intervals with an accuracy of .005%. At the beginning of each new sample set there will be an uncompensated delay of 16 milliseconds while the new sample parameters are accessed.

PROGRAMMER:

Ed Mellinger

DATE:

25 April 1982

Appendix 1

AVRG:

The program AVRG is a modified version of TREC designed for use when noise is a problem in the A/D converter input. AVRG operates in a manner similar to TREC during all modes of operation, with the exception of:

INITIALIZATION. The INTERVAL value should be specified in seconds rather than in tenths of seconds as with TREC. A value of 1 (1 sec) is acceptable to AVRG, but not to TREC.

The maximum total number of samples in all sample sets is 1325 decimal (52D hex).

AQUISITION. The symbol "S" is output once per second while in aquisition mode. After one specified sample interval (1 to 65536 sec) has elapsed, a set of eight samples is aquired over a period of one second and averaged to produce a single value. This value is stored in data memory in the usual fashion.

The following minor irregularities have been noted in the operation of AVRG:

1. The sample interval is actually 1.0023 times the specified sample interval, in seconds.

2. The first sample following a delay period will be erroneous due to the A/D converter turn-on time constant.

2. SOURCE CODE FOR TEMPERATURE RECORDER
a. TREC

AVOCET SYSTEMS 1802 CROSS-ASSEMBLER - VERSION 1.40

SOURCE FILE NAME: TREC.ASM

```
                ;DSDP CORING TOOL TEMPERATURE RECORDER OPERATING SOFTWARE

                ;ED MELLINGER  11 APRIL 1982

                ;UPDATES:      21 APRIL 1982

0000                ;"TREC" IS DMA LOADED INTO LOWEST 600 BYTES OF MEMORY. CPU
                ;RESET CAUSES JUMP TO "INIT" WHICH INITIALIZES CPU REGISTERS AND
                ;PROMPTS FOR SAMPLE-SET PARAMETERS. UP TO 8 SAMPLE SETS MAY BE
                ;SPECIFIED, EACH WITH INDEPENDENT TURN ON DELAY (0-65536 SEC),
                ;SAMPLE INTERVAL (0-6553.6 SEC) AND NUMBER OF SAMPLES (1-1400).
                ;MAX TOTAL NUMBER OF SAMPLES IS 1400. ENTERING A VALUE OF ZERO
                ;SAMPLES TERMINATES "INIT" AND CAUSES "VERIFY" TO BE EXECUTED.
                ;"VERIFY" PRINTS THE SAMPLE-SET PARAMETERS ENTERED DURING "INIT"
                ;AND OUTPUTS AN 8 BIT LINEAR CHECKSUM OVER THE OPERATING PROGRAM
                ;AREA. IT THEN PROMPTS FOR USER ACTION; ENTERING "G" CAUSES
                ;PROGRAM "MAIN" EXECUTION TO BEGIN, ANY OTHER CHARACTER CAUSES A
                ;RESTART AT "INIT".

0000                ;"MAIN" READS THE SAMPLE-SET PARAMETERS STORED IN MEMORY BY
                ;"INIT" AND INITIALIZES THE SAMPLE COUNT, INTERVAL TIMER, AND
                ;DELAY WORKING REGISTERS. "MAIN" THEN JUMPS TO "DLAY" WHICH
                ;DELAYS FOR THE SPECIFIED TIME; DURING DELAY THE TILT SWITCH
                ;OPTION IS ACTIVE AS SPECIFIED BELOW. AFTER DELAY "DLAY" JUMPS TO
                ;"AQUI" WHICH POWERS UP THE A/D CONVERTER AND AQUIRES THE
                ;SPECIFIED NUMBER OF SAMPLES. "AQUI" CALLS SUBROUTINE "TSEC" TO
                ;IMPLEMENT THE SPECIFIED SAMPLE INTERVAL AND "GETDAT" TO STORE
                ;AQUIRED DATA INTO THE DATA MEMORY AREA. WHEN THE SPECIFIED
                ;NUMBER OF SAMPLES HAVE BEEN AQUIRED "AQUI" RETURNS TO "MAIN"
                ;WHERE THE NEXT SET OF SAMPLE-SET PARAMETERS IS READ. IF NUMBER
                ;OF SAMPLES IS ZERO THEN AQUISITION TERMINATES AND THE INSTRUMENT
                ;IS POWERED DOWN WITH AN "INP 7" COMMAND.

0000                ;AFTER INSRUMENT RECOVERY THE CPU IS RESET AND BEGINS
                ;EXECUTION FROM LOCATION 0000. NO-OP INSTRUCTIONS PLACED AT 0000-
                ;0002 BY "MAIN" CAUSE EXECUTION TO TRANSFER TO "PLAYBK" WHICH
                ;OUTPUTS THE STORED DATA AS A SERIES OF RECORDS OF FOUR HEX
                ;CHARACTERS EACH. AFTER THE LAST DATA RECORD IS OUTPUT "PLAYBK"
                ;OUTPUTS THE MESSAGE "STOP" FOLLOWED BY ETX,LF,CR, AND THEN POWERS
                ;DOWN THE RECORDER. DATA MAY BE OUTPUT AGAIN BY RESETTING THE
                ;CPU.

0000                ;THE MAINLINE ROUTINES "INIT", "VERIFY", "MAIN", "DLAY",
                ;"AQUI", AND "PLAYBK" ALL EXECUTE WITH PC=0. THEY CALL THE
                ;UTILITY ROUTINES "INDAT", "OUTDAT", "OUTCHR", "TSEC", "GETDAT",
                ;AND "OUTNUM" TO IMPLEMENT VARIOUS I/O AND TIMING FUNCTIONS. EACH
                ;UTILITY ROUTINE HAS A SEPARATE PC AND IS CALLED BY THE "SEP PC"
                ;TECHNIQUE.
```

```
0000            ;ROUTINES "INIT", "VERIFY", "INDAT", AND "OUTCHR" ARE LOADED
        ;INTO THE DATA AREA OF MEMORY. THEY ARE EXECUTED ONCE DURING
        ;INITIALIZATION AND ARE OVERLAID BY DATA DURING THE ACTUAL
        ;AQUISITION PROCESS. ROUTINES "PLAYBK", "MAIN", "DLAY", "AQUI",
        ;"GETDAT", "TSEC", AND "OUTDAT" ARE LOADED INTO LOW MEMORY AND ARE
        ;ALWAYS PRESENT.
```

```
               ;XXXXXXXXXXXXXXXXXXXX   REGISTER USAGE   XXXXXXXXXXXXXXXXXXXXXX

               ;PROGRAM COUNTERS

               ;MAINPC EQU     R0      MAINLINE ROUTINES PC
               ;TSECPC EQU     R1      "TSEC" PC
               ;INPC   EQU     R3      "INDAT" PC, SHARED WITH "OUTNUM"
               ;OUTDPC EQU     R4      "OUTDAT" PC
               ;GETPC  EQU     R5      "GETDAT" PC
               ;OUTCPC EQU     R6      "OUTCHR" PC
               ;OUTNPC EQU     R3      "OUTNUM" PC, SHARED WITH "INDAT"
               ;DATPTR EQU     R7      DATA MEMORY POINTER, ALSO "LINSUM" POINTER
               ;PARPTR EQU     R8      SAMPLE-SET PARAMETER STRING POINTER
               ;STKPTR EQU     R2      STACK POINTER
               ;ENDFLG EQU     RD      END-OF-DATA-AREA FLAG; SHARED WITH "SMPCTR"

               ;BUFFERS AND WORKING STORAGE

               ;IODAT  EQU     R9      DATA FROM/TO I/O ROUTINES
               ;DLYSTR EQU     RA      DELAY STORAGE FOR "DLAY"
               ;INTSTR EQU     R9      CURRENT SAMPLE INTERVAL FOR "AQUI", SHARED WITH "OUTCHR"

               ;COUNTERS

               ;TSCTR  EQU     RB      "TSEC" INTERVAL COUNTER
               ;DLYCTR EQU     RC      "DLAY" INTERVAL COUNTER
               ;SMPCTR EQU     RD      "AQUI" SAMPLE COUNTER; SHARED WITH "ENDFLG"
               ;TLTCTR EQU     RE      "DLAY" TILT SWITCH COUNTER

               ;SCRATCHPAD

               ;SCRA   EQU     RF      SCRATCHPAD REGISTER


0000           XXXXXXXXXXXXXXXXXXX   EQUATES AND CONSTANTS   XXXXXXXXXXXXXXXXXXXX

               ;I/O PORT NAMES

               ;HBYTE  EQU     INP1    HIGH BYTE OF A/D CONVERTER
               ;LBYTE  EQU     INP2    LOW BYTE OF A/D
               ;ADSTRT EQU     INP3    A/D START SIGNAL
               ;UARTIN EQU     INP4    UART RCVR BUFFER REGISTER
               ;UARTOT EQU     OUT5    UART XMITTER BUFFER REGISTER
               ;UNUSED EQU     INP6    UNUSED I/O SLOT
               ;POWRDN EQU     INP7    POWER DOWN

               ;CONSTANTS

00C4           NOINST  EQU     0C4H    ;OP-CODE OF "NOP" INSTRUCTION
0040           DLAYMK  EQU     040H    ;CHAR "@" FOR "DLAY" TIME MARK
0047           CAPGEE  EQU     'G'     ;USER "GO" SYMBOL
```

TREC
EQUATES

```
 000A         LF      EQU     0AH     ;LINEFEED CHARACTER
 000D         CR      EQU     0DH     ;CARRIAGE RETURN
 0020         SP      EQU     020H    ;SPACE
 0003         ETX     EQU     03H     ;END OF TEXT CHARACTER
```

AVOCET SYSTEMS 1802 CROSS-ASSEMBLER - VERSION 1.40

TREC
PROGRAM

```
                ;*****************************************************************
                ;                          PROGRAM "TREC"
                ;*****************************************************************

0000 C00131           LBR     INIT            ;ON INITIAL CPU RESET,
0003                                          ;JUMP TO "INIT".
0003                                          ;ON LATER RESET,
0003                                          ;JUMP TO "GLITCH".
0003                                          ;ON LAST RESET,
0003                                          ;FALL THRU TO "PLYBK"

                ;************************   PLAYBK   ************************

                ;ENTERED ON CPU RESET AFTER FIRST THREE LOCATIONS ARE FILLED WITH
                ;NOPS BY "MAIN".  USES "PARPTR" AS DATA POINTER, LOADS EACH TWO
                ;BYTE  SAMPLE  INTO "IODAT" AND CALLS "OUTDAT" TO XMIT  DATA  VIA
                ;UART.  EACH 4 DIGIT ASCII HEX VALUE IS FOLLOWED BY LF,CR, AND A
                ;0.2 SEC PAUSE.  RECORDER IS POWERED DOWN AFTER END OF DATA.

0003 E2               SEX     R2              ;FIX STACKPTR AFTER RESET

                ;AT TERMINATION OF "MAIN", "PARPTR" POINTS TO START OF DATA AREA.
                ;TRANSFER IT TO "DATAPTR" FOR USE BY "PLAYBK".

0004 98               GHI     R8
0005 B7               PHI     R7
0006 88               GLO     R8
0007 A7               PLO     R7

                ;COMPARE "DATPTR" TO "ENDFLG" TO CHECK FOR END OF DATA

0008 97         PLAYBK: GHI   R7              ;GET DATA PTR
0009 52               STR     R2              ;PUT ON STACK, NO PUSH
000A 9D               GHI     RD              ;GET FLAG
000B F7               SM                      ;SUBTRACT FLG - PTR
000C 3A14             BNZ     MOVDAT          ;IF .NE. DO DATA OUTPUT

000E 87               GLO     R7              ;ELSE CHECK
000F 52               STR     R2              ;LOW BYTES ALSO
0010 8D               GLO     RD
0011 F7               SM                      ;IF LB'S MATCH,
0012 3223             BZ      PBEND           ;GO END PLAYBACK
0014                                          ;ELSE NEXT DATA PT.

                ;MOVE DATA TO "IODAT", SEND WITH "OUTDAT", SEND LF,CR WITH "OUTCHR"

0014 47         MOVDAT: LDA   R7              ;MOVE 2 BYTES
0015 B9               PHI     R9              ;(1 SAMPLE) INTO
0016 47               LDA     R7              ;I/O BUFFER
0017 A9               PLO     R9              ;AND SEND WITH
0018 D4               SEP     R4              ;CALL "OUTDAT"
```

```
0019 D6                  SEP    R6              ;CALL "OUTCHR" TO
001A 0A0D00              DB     LF,CR,0         ;SEND LF,CR

001D F802                LDI    2               ;DELAY 0.2 SEC
001F AB                  PLO    RB              ;BY CALL TO
0020 D1                  SEP    R1              ;"TSEC"

0021 3008                BR     PLAYBK          ;KEEP SENDING DATA

0023 D6         PBEND:   SEP    R6              ;SEND MSG. WHEN THRU
0024 53544F50            DB     'STOP'
0028 030A0D00            DB     ETX,LF,CR,0

002C 6F                  INP7                   ;THEN POWER DOWN
002D 00                  IDL                    ;AND IDLE FOR SAFETY

002E            XXXXXXXXXXXXXXXXXXXXXXX   OUTDAT/OUTNUM   XXXXXXXXXXXXXXXXXXXXXXX

                ;"OUTDAT" TRANSMITS CONTENTS OF REGISTER "IODAT" AS 4 ASCII HEX DIGITS.
                ;"OUTDAT" ENTERED VIA "SEP OUTDPC" AND EXITS VIA "SEP MAINPC".

002E D0         OTEXIT:  SEP    R0              ;RETURN TO CALLER

002F 99         OUTDAT:  GHI    R9              ;HIGH BYTE TO XMIT
0030 F6                  SHR
0031 F6                  SHR
0032 F6                  SHR
0033 F6                  SHR                    ;ISOLATE HIGH NIBBLE
0034 D3                  SEP    R3              ;SEND IT

0035 99                  GHI    R9              ;HIGH BYTE
0036 FA0F                ANI    0FH             ;LOW NIBBLE
0038 D3                  SEP    R3              ;SEND IT

0039 89                  GLO    R9              ;LOW BYTE
003A F6                  SHR
003B F6                  SHR
003C F6                  SHR
003D F6                  SHR                    ;HIGH NIBBLE
003E D3                  SEP    R3              ;SEND IT

003F 89                  GLO    R9              ;LOW BYTE
0040 FA0F                ANI    0FH             ;LOW NIBBLE
0042 D3                  SEP    R3              ;SEND IT

0043 302E                BR     OTEXIT          ;RESET "OUTDPC" BEFORE EXIT

                ;"OUTNUM" -- DEDICATED SUBROUTINE FOR "OUTDAT".  CONVERTS 4 BIT BINARY
                ;VALUE IN D REG TO ASCII HEX (0-9, A-F) AND XMITS VIA UART.
```

```
 0045 D4        ONEXIT: SEP     R4              ;RETURN TO "OUTDAT"

 0046 FF0A      OUTNUM: SMI     0AH             ;TEST FOR <= 9
 0048 C7                LSNF                    ;SKIP ADD IF YES
 0049 FC07              ADI     7               ;IF A-F, ADD EXTRA

 004B FC3A              ADI     03AH            ;ADD NET 30 OR
 004D                                           ;NET 36 TO GET
 004D                                           ;ASCII CHARACTER.
 004D 52                STR     R2              ;PUT ON STACK, NO PUSH
 004E 344E      ONLOOP: B1      ONLOOP          ;WAIT FOR UART TRE
 0050 65                OUT5                    ;SEND TO UART
 0051 22                DEC     R2              ;UNDO AUTOINCREMENT
 0052 3045              BR      ONEXIT          ;RESET "OUTNPC"

                ;XXXXXXXXXXXXXXXXXXXX   MAIN/DLAY/AQUI   XXXXXXXXXXXXXXXXXXXXXX

                ;"MAIN" READS SAMPLE-SET PARAMETERS FROM MEMORY (POINTED TO BY
                ;(PARPTR") AND STORES THEM IN "SMPCTR", "INTSTR", AND "DLYSTR".
                ;IF SPECIFIED NUMBER OF SAMPLES IS ZERO, AQUISITION STOPS AND THE
                ;RECORDER IS POWERED DOWN.  IF # SAMPLES IS NONZERO "MAIN" JUMPS
                ;TO "DLAY" FOR DELAY BEFORE AQUISITION BEGINS.  AFTER DELAY,
                ;"DLAY" JUMPS TO "AQUI" WHERE THE SPECIFIED NUMBER OF SAMPLES ARE
                ;AQUIRED, FOLLOWED BY A RETURN TO "MAIN" WHERE THE NEXT SAMPLE-
                ;SET'S PARAMETERS ARE READ.

 0054                   ;MAIN, DLAY, AND AQUI EXECUTE WITH PC=MAINPC=R0.  IN
                ;GENERAL, THETIMING OF THESE ROUTINES IS CRITICAL TO OVERALL
                ;TIMING ACCURACY.  ANY PROGRAM CHANGES SHOULD THEREFORE BE MADE
                ;WITH CARE.

 0054 48        MAIN:   LDA     R8              ;GET SAMPLE HB
 0055 BD                PHI     RD              ;STORE
 0056 52                STR     R2              ;ALSO SAVE TEMP.
 0057 48                LDA     R8              ;GET # SAMPLES LB
 0058 AD                PLO     RD              ;STORE
 0059 F1                OR                      ;MERGE LB AND HB
 005A 3A6C               BNZ     NOIDLE          ;KEEP GOING IF<>0
 005C                                           ;ELSE POWER DOWN

                ;BEFORE POWERING DOWN PUT "NO-OPS" AT 0000-0002, TO ALLOW
                ;"PLAYBK" TO BE EXECUTED AFTER NEXT CPU RESET.  (NOTE D=0 HERE)

 005C BF                PHI     RF              ;FIRST NEED
 005D AF                PLO     RF              ;POINTER TO 0000

 005E F8C4              LDI     NOINST          ;GET "NOP" OPCODE
 0060 5F                STR     RF              ;PUT AT 0000
 0061 1F                INC     RF
 0062 5F                STR     RF              ;AND AT 0001
 0063 1F                INC     RF
```

```
 0064 5F               STR     RF              ;AND AT 0002

             ;ALSO SAVE LAST DATA LOCATION AS FLAG FOR "PLAYBK"

 0065 97               GHI     R7
 0066 BD               PHI     RD
 0067 87               GLO     R7
 0068 AD               PLO     RD

 0069 7A               REQ                     ;TURN OFF A/D
 006A 6F               INP7                    ;POWER DOWN
 006B 00               IDL                     ;IDLE TO BE SURE

             ;IF # SAMPLES <> 0 GET INTERVAL AND DELAY PARAMETERS ALSO

 006C 48      NOIDLE:  LDA     R8              ;GET AND STORE
 006D B9               PHI     R9              ;SAMPLE INTERVAL
 006E 48               LDA     R8              ;HB THEN LB.
 006F A9               PLO     R9

 0070 48               LDA     R8              ;GET AND STORE
 0071 BA               PHI     RA              ;DELAY TIME
 0072 48               LDA     R8              ;HB THEN LB.
 0073 AA               PLO     RA

             ;FALL  THRU INTO "DLAY" WHEN ALL PARAMETERS ARE  LOADED.  "DLAY"
             ;MOVES "DLYSTR" INTO "DLYCTR", DECREMENTS "DLYCTR" ONCE PER
             ;SECOND UNTIL "DLYCTR" IS ZERO, THEN JUMPS TO "AQUI".  TILT
             ;SWITCH IS SAMPLED EVERY SECOND DURING DELAY.  IF SWITCH OPENS
             ;FOR 60 CONSECUTIVE SAMPLES (1 MINUTE) THEN "RESTRT" IS EXECUTED.
             ;"RESTRT" RESETS DELAY COUNTER TO ITS INITIAL VALUE ("DLYSTR"
             ;INTO "DLYCTR"), THEN SAMPLES TILT SWITCH EVERY 4 SECONDS.  IF
             ;SWITCH CLOSES FOR 75 CONSECUTIVE SAMPLES (5 MINUTES) THEN "DLAY"
             ;IS EXECUTED AND DELAY STARTS OVER.

             ;TIMING IN THESE ROUTINES IS CRITICAL; MAKE CHANGES CAREFULLY.

 0074 9A      DLAY:    GHI     RA              ;GET DELAY TIME
 0075 BC               PHI     RC              ;FROM STORAGE
 0076 8A               GLO     RA              ;INTO
 0077 AC               PLO     RC              ;WORKING COUNTER.

 0078 F83C             LDI     03CH            ;60 SEC TIMEOUT
 007A AE               PLO     RE              ;TO DETECT TILT.

             ;("GLITCH" IS THE RE-ENTRY POINT IN CASE OF A SPURIOUS CPU RESET
             ;DURING "DLAY" OR "AQUI"; IT IS ENTERED AFTER CPU RESET VIA A
             ;LONG BRANCH PLACED AT 0000-0002 BY "INIT".  RECOVERY FROM
             ;SPURIOUS RESETS IS POSSIBLE DURING "DLAY" AND "AQUI", WITH
             ;POSSIBLE LOSS OF ONE DELAY OF AQUISITION CYCLE.  GLITCHES DURING
             ;"MAIN" ARE FATAL SINCE "PARPTR" WILL BE INCORRECTLY POSITIONED
```

```
                ;UPON RE-ENTRY.)

 007B E2        GLITCH: SEX     R2              ;FIX R(X) AFTER RESET

                ;TEST FOR DELAY FINISHED; TEST FOR TILT

 007C 9C        DYLOOP: GHI     RC              ;TEST DELAY
 007D 52                STR     R2              ;BY MERGING
 007E 8C                GLO     RC              ;"DLYCTR" HB+LB.
 007F F1                OR                      ;IF BOTH ZERO
 0080 32B4              BZ      AQUI            ;GO DO AQUI.

 0082 8E                GLO     RE              ;ELSE TEST FOR
 0083 32A2              BZ      RESTRT          ;TILY, RESTART IF YES

                ;OUTPUT "@" FOR TIMING, DELAY ONE SECOND, CHECK TILT SWITCH

 0085                                           ;IF ALL OK,
 0085 6E                INP6                    ;DO TIME MARKS
 0086 F840              LDI     DLAYMK          ;XMIT MARK VIA
 0088 52                STR     R2              ;UART EVERY SEC.
 0089 65                OUT5
 008A 22                DEC     R2              ;UNDO AUTOINCR.

 008B 2C                DEC     RC              ;DEC DELAY COUNTER REG

 008C F809              LDI     9               ;COARSE DELAY,
 008E AB                PLO     RB              ;0.9 SEC VAI
 008F D1                SEP     R1              ;"TSEC" CALL.

 0090 F849              LDI     049H            ;FINE DELAY (49H=73D)
 0092 FF01      DELOOP: SMI     1               ;TO FILL OUT DYLOOP
 0094 7A                REQ                     ;(ALSO OFF A/D)
 0095 3A92              BNZ     DELOOP          ;TO EXACT 1.0 SEC

 0097 3F9E              BN4     DTILT           ;CHECK TILT SW,
 0099 F83C              LDI     03CH            ;IF NO TILT,
 009B AE                PLO     RE              ;RESET TILT COUNTER
 009C 307C              BR      DYLOOP          ;KEEP DELAYING.

 009E 2E        DTILT:  DEC     RE              ;IF TILT, DEC COUNTER.
 009F 7A                REQ                     ;(2 CYCLE NO-OP)
 00A0 307C              BR      DYLOOP          ;KEEP DELAYING.

                ;"RESTRT"  RESETS THE DELAY COUNTER ("DLYCTR") AND WAITS FOR  THE
                ;TILT SWITCH TO OPEN AND STAY OPEN 5 MINUTES.

 00A2 F84B      RESTRT: LDI     04BH            ;4BH=75D FOR
 00A4 AE                PLO     RE              ;5 MIN RESTART DELAY.

 00A5 8E        RSLOOP: GLO     RE              ;TEST FOR UNTILT
```

```
 00A6 3274             BZ      DLAY            ;RE-START DELAY IF YES

 00A8 F828             LDI     028H            ;4.0 SECOND
 00AA AB               PLO     RB              ;COARSE DELAY ONLY
 00AB D1               SEP     R1              ;VIA "TSEC" CALL.

 00AC 37B1             B4      NOTILT          ;TEST TILT SW
 00AE F84C             LDI     04CH            ;IF OPEN, RESET
 00B0 AE               PLO     RE              ;TILT COUNTER (+1)

 00B1 2E      NOTILT:  DEC     RE              ;IF CLOSED, DEC
 00B2 30A5             BR      RSLOOP          ;COUNTER AND LOOP.

              ;"AQUI" IS ENTERED FROM "DLAY". "AQUI" CHECKS SAMPLE COUNTER
              ;REG. "SMPCTR" FOR ZERO, RETURNS TO MAIN IF YES. IF NO, "AQUI"
              ;POWERS UP A/D, INITIALIZES "TSEC" COUNTER WITH SAMPLE INTERVAL
              ;AND CALLS "TSEC". AFTER "TSEC" TIMES THE SPECIFIED INTERVAL,
              ;"AQUI" STARTS A/D CONVERSION AND CALLS "GETDAT" TO TRANSFER
              ;CONVERTER DATA TO MEMORY IN DESIFED FORMAT (PACKED OR UNPACKED).
              ;AFTER TRANSFER, "AQUI" CHECKS DATA POINTER "DATPTR" AND PREVENTS
              ;IT FROM INCREMENTING BEYOND THE END OF DATA MEMORY (LOCATION
              ;0BFF).

              ;"AQUI" IS A MAINLINE ROUTINE (PC=MAINPC=R0). ITS TIMING IS
              ;CRITICAL AND MODIFICATIONS SHOULD BE MADE WITH CARE.

              ;FIRST CHECK FOR DONE

 00B4 9D      AQUI:    GHI     RD              ;GET SAMPLE COUNTER HB
 00B5 52               STR     R2              ;PUT ON STACK, NO PUSH
 00B6 8D               GLO     RD              ;GET SAMPLE CTR LB
 00B7 F1               OR                      ;MERGE HB+LB
 00B8 3254             BZ      MAIN            ;IF BOTH ZERO, GO DO
 00BA                                          ;NEXT SAMPLE SET.

              ;IF NOT DONE, SEND MARK AND WAIT 1 SAMPLE INTERVAL

 00BA 7B               SEQ                     ;PWR UP A/D
 00BB 6E               INP6                    ;PULSE I/O 6 LINE
 00BC 2D               DEC     RD              ;DEC SAMPLE COUNTER
 00BD 99               GHI     R9              ;MOVE INTERVAL
 00BE BB               PHI     RB              ;INTO "TSEC" WRKING
 00BF 89               GLO     R9              ;COUNTER
 00C0 AB               PLO     RB
 00C1 2B               DEC     RB              ;LESS 0.1 SEC FOR
 00C2                                          ;LATER FINE DELAY.
 00C2 D1               SEP     R1              ;DELAY SPEC'D TIME
 00C3 6B               INP3                    ;START A/D CONVERSION

 00C4 F86A             LDI     06AH            ;FINE DELAY
 00C6 FF01    AQLOOP:  SMI     1               ;WAIT FOR A/D
```

```
 00C8 3AC6          BNZ     AQLOOP

          ;GET DATA FROM A/D AND CHECK FOR DATA MEMORY FULL

 00CA D5            SEP     R5              ;CALL "GETDAT" TO
 00CB                                       ;MOVE DATA TO MEM

 00CB 97            GHI     R7              ;CHECK FOR
 00CC FF0C          SMI     0CH             ;DATA MEM FULL
 00CE 3BB4          BM      AQUI            ;LOOP IF NO

 00D0 27            DEC     R7              ;IF YES,
 00D1 27            DEC     R7              ;DEC POINTER
 00D2 3054          BR      MAIN            ;AND FORCE NEXT
          ;(THIS ALLOWS CORRECT TERMINATION OF AQUISITION IN "MAIN".  ALL
          ;SAMPLE PARAMETERS MUST BE READ TO CORRECTLY POSITION "PARPTR"
          ;FOR PLAYBACK.)

          ;"GETDAT" -- NONPACKING VERSION.  GETS A/D DATA AND STORES
          ;SEQUENTIALLY IN MEMORY.  (PACKING VERSION ALLOWS 33% MORE
          ;SAMPLES IF 12-BIT SAMPLES ARE USED).

 00D4 D0          GDEXIT: SEP     R0        ;RETURN TO "AQUI"
 00D5 69          GETDAT: INP1              ;GET A/D HB
 00D6 57                  STR     R7        ;PUT IN DATA MEM
 00D7 17                  INC     R7        ;INC POINTER
 00D8 6A                  INP2              ;GET A/D LB
 00D9 57                  STR     R7        ;PUT IN DATA MEM
 00DA 17                  INC     R7        ;INC POINTER

 00DB 30D4                BR      GDEXIT    ;RESTORE "GETPC"

          ;xxxxxxxxxxxxxxxxxxxxxxxxx   OUTCHR   xxxxxxxxxxxxxxxxxxxxxxxxxx

 00DD                 ;ENTERED VIA "SEP OUTCPC".  OUTPUTS CHARACTER STRING
          ;POINTED TO BY "MAINPC" UNTIL A 00 BYTE IS ENCOUNTERED, THEN
          ;RETURNS VIA "SEP MAINPC".  ALTERS M(STKPTR).

 00DD D0          OCEXIT: SEP     R0        ;RETURN TO CALLER
 00DE 40          OUTCHR: LDA     R0        ;GET CHARACTER
 00DF 32DD                BZ      OCEXIT    ;IF 00, EXIT
 00E1 52                  STR     R2        ;ELSE PUT ON STACK
 00E2 34E2        OCLOOP: B1      OCLOOP    ;WAIT FOR UART
 00E4 65                  OUT5              ;XMIT WHEN READY
 00E5 22                  DEC     R2        ;UNDO AUTOINCR.
 00E6 30DE                BR      OUTCHR    ;GET NEXT CHAR

          ;xxxxxxxxxxxxxxxxxxxxxxxxx   TSEC   xxxxxxxxxxxxxxxxxxxxxxxxxx

          ;"TSEC" IS A 0.1 SEC TIMING ROUTINE.  TIME FROM BEGINNING OF
          ;EXECUTION OF FIRST INSTRUCTION (DEC TSCTR) TO END OF EXECUTION
```

AVOCET SYSTEMS 1802 CROSS-ASSEMBLER - VERSION 1.40

TREC
PROGRAM

```
                ;OF LAST INSTRUCTION (SEP MAINPC) IS
00E8                    ;(0.1) X (VALUE IN "TSCTR") SECONDS,
                ;ASSUMING A 38.400 KHZ CPU CLOCK.
                ;CALL WITH "SEP TSECPC".  RETURN TO MAINPC (R0).

00E8 D0         TSEXIT: SEP     R0              ;RETURN TO CALLER
00E9 2B         TSEC    DEC     RB              ;DEC COUNTER
00EA F874               LDI     074H            ;SET UP LOCAL LOOP

00EC FF01       TSLOOP: SMI     1
00EE 3AEC               BNZ     TSLOOP

00F0 9B                 GHI     RB              ;CHECK COUNTER
00F1 52                 STR     R2              ;PUT HB ON STACK
00F2 8B                 GLO     RB              ;GET LB
00F3 F1                 OR                      ;MERGE HB+LB
00F4 32E8               BZ      TSEXIT          ;EXIT IF BOTH 0

00F6 30E9       ENDCHK: BR      TSEC            ;ELSE KEEP LOOPING
                ;("ENDCHK"  IS  LABEL  FOR  LAST  LOCATION  TO  CHECKSUM  DURING
                ;"VERIFY"

                ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                ;XXXXXXXXXXXXXXXXXX   START OF DATA AREA   XXXXXXXXXXXXXXXXXXX
                ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

0100                    ORG     $+8             ;SMALL STACK AREA
0101            STKTOP: ORG     $+1
0131            PARAM:  ORG     $+48            ;SPACE FOR
0131                                            ;8 SAMPLE-SET PARAMETER STRINGS

                ;EACH SAMPLE-SET PARAMETER STRING IS 6 BYTES LONG;
                ;TWO BYTES EACH (HB FIRST) FOR:  # SAMPLES,  SAMPLE INTERVAL, AND
                ;TURN-ON  DELAY.  DURING AQUISITION,  SATA IS STORED  IN  MEMORY
                ;BEGINNING  DIRECTLY AFTER (ABOVE) THE LAST PARAMETER.   THE LAST
                ;PARAMETER  IS  ALWAYS THE "ZERO SAMPLES" FLAG  WHICH  TERMINATES
                ;INPUT AND AQUISITION.

                ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                ;XXXXXXXXXXXXXXX   INITIALIZATION ROUTINE AREA   XXXXXXXXXXXXXXX
                ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

                ;THE  FOLLOWING  ROUTINES ARE LOADED INTO THE DATA AREA  AND  ARE
                ;EXECUTED   ONCE  DURING  INITIALIZATION.   THEY  ARE   OVERLAID
                ;(DESTROYED) BY DATA DURING THE ACTUAL AQUISITION PROCESS.

                ;XXXXXXXXXXXXXXXXXXXXXXXXXX   INIT   XXXXXXXXXXXXXXXXXXXXXXXXXXX

0131                    ;ENTERED VIA LONG BRANCH INSTRUCTION AT 0000-0002,  AFTER
                ;CPU RESET.   INITIALIZES REGISTERS TSECPC,  INPC, OUTPC, OUTCPC,
                ;GETPC,  STKPTR,  PARPTR; SETS X=STKPTR; POWERS DOWN A/D. PROMPTS
```

```
                    ;USER FOR INPUT OF SAMPLE-SET PARAMETERS:
                    ;  #SAMPLES?  # OF SAMPLES IN THIS SET; ZERO STOPS FURHTER INPUT.
                    ;  INTERVAL?  SAMPLE INTERVAL FOR THIS SET, IN TENTHS OF SECONDS.
                    ;     DELAY?  TURN ON DELAY BEFORE AQUIRING THIS SET, IN SECONDS.

                    ;THE  RESPONSE  TO EACH PROMPT SHOULD BE A ONE TO FOUR  CHARACTER
                    ;HEX INTEGER FOLLOWED BY A CARRAIGE RETURN.   THE INPUT  PROTOCOL
                    ;IS  UT4  FORMAT,  I.E.  ONLY THE LAST 4 DIGITS ARE  SIGNIFICANT;
                    ;THUS  ERRORS  MAY  BE CORRECTED BY TYPING UNTIL  THE  NUMBER  IS
                    ;CORRECT  AND  THEN TYPING  "RETURN".   ILLEGAL  OR  UNRECOGNIZED
                    ;CHARACTERS ARE NOT ECHOED BACK TO THE USER INPUT DEVICE.

0131                        ;"INIT"  CONTINUES  TO PROMPT FOR  SAMPLE-SET  PARAMETERS
                    ;UNTIL  A  VALUE  0  IS READ FOR # SAMPLES.   UP  TO  8  SETS  OF
                    ;PARAMETERS  MAY  BE ENTERED;  MORE WILL OVERLAY  PROGRAM  AREAS.
                    ;"INIT" EXITS WITH A JUMP TO "VERIFY", KEEPING PC=MAINPC=R0.


0131 7A             INIT:   REQ                     ;TURN OFF A/D

                    ;INITIALIZE WORKING REGISTERS AS REQ'D

0132 F800                   LDI     A.1(TSEC)       ;INITIALIZE
0134 B1                     PHI     R1              ;"TSEC" PC.
0135 F8E9                   LDI     A.0(TSEC)
0137 A1                     PLO     R1

0138 F802                   LDI     A.1(INDAT)
013A B3                     PHI     R3              ;"INDAT" PC.
013B F85A                   LDI     A.0(INDAT)
013D A3                     PLO     R3

013E F800                   LDI     A.1(OUTDAT)
0140 B4                     PHI     R4              ;"OUTDAT" PC.
0141 F82F                   LDI     A.0(OUTDAT)
0143 A4                     PLO     R4

0144 F800                   LDI     A.1(OUTCHR)
0146 B6                     PHI     R6              ;"OUTCHR" PC.
0147 F8DE                   LDI     A.0(OUTCHR)
0149 A6                     PLO     R6

014A F800                   LDI     A.1(GETDAT)
014C B5                     PHI     R5              ;"GETDAT" PC.
014D F8D5                   LDI     A.0(GETDAT)
014F A5                     PLO     R5

0150 F801                   LDI     A.1(STKTOP)
0152 B2                     PHI     R2              ;STACK POINTER
0153 F800                   LDI     A.0(STKTOP)
0155 A2                     PLO     R2
```

```
 0156 F801            LDI    A.1(PARAM)      ;PARAMETER
 0158 B8              PHI    R8              ;STRING POINTER
 0159 F801            LDI    A.0(PARAM)
 015B A8              PLO    R8

 015C E2              SEX    R2              ;X=STKPTR ALWAYS

            ;PROMPT USER FOR SAMPLE-SET PARAMETERS

 015D D6      PROMPT: SEP    R6              ;CALL "OUTCHR" TO
 015E 0A0D00          DB     LF,CR,0         ;SEND LF,CR.

 0161 F800            LDI    0               ;SET UP 0.2 SEC
 0163 BB              PHI    RB              ;DELAY AFTER CR
 0164 F802            LDI    2
 0166 AB              PLO    RB
 0167 D1              SEP    R1              ;CALL "TSEC"

            ;PROMPT AND GET # OF SAMPLES

 0168 D6              SEP    R6              ;CALL "OUTCHR"
 0169 2353414D        DB     '#SAMPLES'
 0171 3F200300        DB     '?',SP,ETX,0

 0175 D3              SEP    R3              ;CALL "INDAT"
 0176 99              GHI    R9              ;GET RETURN HB
 0177 58              STR    R8              ;STORE IN MEMORY
 0178 52              STR    R2              ;ALSO ON STACK
 0179 18              INC    R8              ;INC MEM PTR

 017A 89              GLO    R9              ;GET RETURN LB
 017B 58              STR    R8              ;STORE IN MEMORY
 017C 18              INC    R8              ;INC MEM PTR
 017D F1              OR                     ;MERGE HB AND LB
 017E 32AD            BZ     BURN            ;IF 0, NO MORE INPUT.
 0180                                        ;ELSE GET INTVL, DELAY.

            ;PROMPT AND GET SAMPLE INTERVAL

 0180 D6              SEP    R6              ;SEND PROMPT
 0181 202020          DB     SP,SP,SP
 0184 494E5445        DB     'INTERVAL'
 018C 3F200300        DB     '?',SP,ETX,0

 0190 D3              SEP    R3              ;GET INPUT
 0191 99              GHI    R9
 0192 58              STR    R8
 0193 18              INC    R8
 0194 89              GLO    R9
 0195 58              STR    R8
```

```
 0196 18                INC     R8

              ;PROMPT AND GET TURN ON DELAY TIME (DELAY BEFORE SAMPLING STARTS)

 0197 D6                SEP     R6              ;SEND PROMPT
 0198 202020            DB      SP,SP,SP
 019B 44454C41          DB      'DELAY?'
 01A1 200300            DB      SP,ETX,0

 01A4 D3                SEP     R3              ;GET INPUT
 01A5 99                GHI     R9              ;STORE IN
 01A6 58                STR     R8              ;MEMORY
 01A7 18                INC     R8              ;PARAM STRING AREA
 01A8 89                GLO     R9
 01A9 58                STR     R8
 01AA 18                INC     R8

 01AB 305D              BR      PROMPT          ;GET NEXT
 01AD                                           ;PARAMETER SET

              ;CHANGE "LONG BRANCH" INSTRUCTION AT 0000-0002 TO RE-ENTER "DLAY"
              ;IN CASE OF SPURIOUS RESET.

 01AD F800    BURN:     LDI     0               ;SET UP POINTER
 01AF BF                PHI     RF              ;TO LOCATION
 01B0 AF                PLO     RF
 01B1 1F                INC     RF              ;OF JUMP ADDRESS

 01B2 F800              LDI     A.1(GLITCH)     ;GET NEW ADDRESS
 01B4 5F                STR     RF              ;STORE IN 0001
 01B5 1F                INC     RF
 01B6 F87B              LDI     A.0(GLITCH)
 01B8 5F                STR     RF              ;AND IN 0002

              ;FALL THRU TO VERIFY WHEN THRU

              ;XXXXXXXXXXXXXXXXXXXXXXXX    VERIFY   XXXXXXXXXXXXXXXXXXXXXXXXXX

 01B9                   ;OUTPUTS # SAMPLES, INTERVAL, AND DELAY DATA FOR USER
              ;INSPECTION.  COMPUTES LINEAR CHECKSUM (LINEAR SUM OF MEMORY
              ;BYTES) OVER PROGRAM AREA FROM 0000 TO LABEL "ENDCHK" AT END OF
              ;"TSEC", AND OUTPUTS RESULT.  PROMPTS WITH "ACTION...?" FOR USER
              ;RESPONSE.  ENTER UPPERCASE "G" TO BEGIN EXECUTION OF "MAIN", ANY
              ;OTHER CHARACTER TO START "INIT" OVER AGAIN.  "PARPTR" AND
              ;"DATPTR" ARE RE-INITIALIZED BEFORE THE JUMP TO "MAIN".

 01B9                   ;"VERIFY" EXECUTES WITH PC=MAINPC=R0.

 01B9 F800    VERIFY:   LDI     A.1(OUTNUM)     ;FIRST BUSINESS IS
 01BB B3                PHI     R3              ;TO INIT "OUTNPC"
 01BC F846              LDI     A.0(OUTNUM)     ;NOW THAT "INDAT"
```

```
 01BE A3              PLO    R3              ;IS FINISHED

 01BF F801            LDI    A.1(PARAM)      ;RESET PARPTR TO
 01C1 B8              PHI    R8              ;START OF
 01C2 F801            LDI    A.0(PARAM)      ;PARAM STRING
 01C4 A8              PLO    R8              ;BEFORE VERIFYING

 01C5 D6              SEP    R6              ;CALL OUTCHR AND
 01C6 0A0A00          DB     LF,LF,0         ;LEAVE TWO BLANK LINES

 01C9 D6      VLOOP:  SEP    R6              ;VLOOP IS
 01CA 0A0D00          DB     LF,CR,0         ;OUTPUT LOOP

 01CD F802            LDI    2               ;0.2 SEC DELAY
 01CF AB              PLO    RB              ;AFTER CR, VIA
 01D0 D1              SEP    R1              ;CALL TO "TSEC"

              ;OUTPUT #SAMPLES; EXIT VLOOP IF =0.

 01D1 48              LDA    R8              ;GET #SAMPLES HB
 01D2 B9              PHI    R9              ;PUT IN OUT BUFFER
 01D3 52              STR    R2              ;ALSO ON STACK
 01D4 48              LDA    R8              ;GET #SAMPLES LB
 01D5 A9              PLO    R9              ;PUT IN OUT BUFFER
 01D6 F1              OR                     ;MERGE HB AND LB
 01D7 C2020F          LBZ    LINSUM          ;EXIT VLOOP IF 0

 01DA D6              SEP    R6              ;ELSE SEND TEXT
 01DB 2353414D        DB     '#SAMPLES'
 01E3 3D2000          DB     '=',SP,0
 01E6 D4              SEP    R4              ;THEN SEND DATA

              ;OUTPUT SAMPLE INTERVAL

 01E7 48              LDA    R8              ;GET INTERVAL HB
 01E8 B9              PHI    R9              ;PUT IN BUFFER
 01E9 48              LDA    R8              ;GET INTERVAL LB
 01EA A9              PLO    R9              ;PUT IN BUFFER

 01EB D6              SEP    R6              ;SEND TEXT
 01EC 202020          DB     SP,SP,SP
 01EF 494E5445        DB     'INTERVAL'
 01F7 3D2000          DB     '=',SP,0
 01FA D4              SEP    R4              ;THEN SEND DATA

              ;OUTPUT TURN-ON DELAY

 01FB 48              LDA    R8              ;GET HB
 01FC B9              PHI    R9
 01FD 48              LDA    R8              ;GET LB
 01FE A9              PLO    R9
```

AVOCET SYSTEMS 1802 CROSS-ASSEMBLER - VERSION 1.40

TREC
PROGRAM

```
01FF D6                SEP   R6              ;SEND TEXT
0200 202020            DB    SP,SP,SP
0203 44454C41          DB    'DELAY=',SP,0
020B D4                SEP   R4              ;THEN SEND DATA
020C C001C9            LBR   VLOOP           ;REPEAT FOR NEXT
020F                                         ;PARAMETER SET

            ;COMPUTE CHECKSUM AND OUTPUT RESULT.

020F F800     LINSUM: LDI   0               ;FIRST SET UP
0211 B7               PHI   R7              ;POINTER TO
0212 A7               PLO   R7              ;PROGRAM AREA

0213 B9               PHI   R9              ;CLEAR BUFFER HB
0214 E7               SEX   R7              ;READY FOR ADDING

0215 89       CKLOOP: GLO   R9              ;GET OLD SUM
0216 F4               ADD                   ;ADD MEMORY BYTE
0217 A9               PLO   R9              ;SAVE NEW SUM
0218 17               INC   R7              ;INC POINTER TO NEXT BYTE.

0219 97               GHI   R7              ;TEST IF THRU BY
021A FF00             SMI   A.1(ENDCHK)     ;COMPARING PTR
021C CB0215           LBNF  CKLOOP          ;AND FLAG LABEL.
021F                                        ;KEEP LOOPING IF NO
021F 87               GLO   R7              ;BOTH BYTES
0220 FFF6             SMI   A.0(ENDCHK)     ;MUST MATCH
0222 CB0215           LBNF  CKLOOP          ;TO BE DONE

0225 E2               SEX   R2              ;RE-ENABLE STACK
0226 D6               SEP   R6              ;WHEN DONE, SEND
0227 20202043         DB    SP,SP,SP,'CHECK'
022F 53554D3D         DB    'SUM=',SP,0
0235 D4               SEP   R4              ;AND SEND DATA

            ;OUTPUT PROMPT AND READ GO/NO GO CHARACTER.

0236 D6               SEP   R6              ;SEND PROMPT
0237 20202041         DB    SP,SP,SP,'ACT'
023D 494F4E3F         DB    'ION?',SP,ETX,0

0244 3544     ACLOOP: B2    ACLOOP          ;AWAIT UART REPLY
0246 6C               INP4                  ;GET REPLY
0247 FF47             SMI   CAPGEE          ;IS IT "G" ?
0249 CA0131           LBNZ  INIT            ;IF NO, "INIT"
024C                                        ;IF YES, "MAIN"

            ;AFTER LAST SAMPLE-SET PARAMETER HAS BEEN READ "PARPTR" POINTS TO
            ;START OF DATA AREA.   TRANSFER IT OT "DATPTR", THE RE-INITIALIZE
            ;"PARPTR".  THEN JUMP TO "MAIN" TO BEGIN AQUISITION PROCESS.
```

```
 024C 98              GHI     R8
 024D B7              PHI     R7
 024E 88              GLO     R8
 024F A7              PLO     R7

 0250 F801            LDI     A.1(PARAM)
 0252 B8              PHI     R8
 0253 F801            LDI     A.0(PARAM)
 0255 A8              PLO     R8
 0256 C00054          LBR     MAIN                ;END OF OPERATOR
 0259                                             ;INITIALIZATION

              ;XXXXXXXXXXXXXXXXXXXXXXXX   INDAT   XXXXXXXXXXXXXXXXXXXXXXXX

 0259                 ;CALLED VIA "SEP INPC".  CLEARS "IODAT", THEN READS RCVD.
              ;CHARACTERS  FROM  UART.   HEX DIGITS (0-9,A-F) ARE CONVERTED  TO
              ;4  BIT VALUES AND SHIFTED INTO "IODAT" FROM RIGHT TO  LEFT,  AND
              ;ALSO  ECHOED TO USER.   NON-HEX CHARACTERS ARE  IGNORED.   INPUT
              ;STRING IS TERMINATED WITH A CR; RETURN IS VIA "SEP MAINPC".
              ;ASSUMES X=STKPTR.

 0259 D0      INEXIT: SEP     R0                  ;RETURN TO CALLER
 025A F800    INDAT:  LDI     0                   ;ON ENTRY
 025C B9              PHI     R9                  ;CLEAR BUFFER
 025D A9              PLO     R9

              ;INPUT A CHARACTER, CHECK IT, CONVERT IT TO 4 BITS, ECHO IT

 025E 355E    ILOOP:  B2      ILOOP               ;WAIT FOR CHARACTER
 0260 6C              INP4                        ;THEN GET IT

 0261 FF0D            SMI     CR                  ;CHECK FOR CR
 0263 3259            BZ      INEXIT              ;EXIT IF YES

 0265 F0              LDX                         ;RE-GET CHAR
 0266 FF30            SMI     030H                ;TEST FOR NON-HEX
 0268 3B5E            BM      ILOOP               ;IF NON, TRY AGAIN

 026A FF0A            SMI     0AH                 ;ELSE TEST FOR
 026C 3B78            BM      DECIML              ;DECIMAL, GO
 026E                                             ;CONVERT IF YES
 026E FF07            SMI     7                   ;ELSE TEST #3A-40
 0270 3B5E            BM      ILOOP               ;TRY AGAIN IF YES
 0272 FF06            SMI     6                   ;ELSE TEST >= #46
 0274 335E            BPZ     ILOOP               ;TRY AGAIN IF YES

 0276 FC06            ADI     6                   ;ELSE OK HEX A-F
 0278 FC0A    DECIML: ADI     0AH                 ;MAKE 4 BIT VALUE
 027A AF              PLO     RF                  ;SAVE TEMP
```

```
 027B 347B      INLOOP: B1      INLOOP          ;WAIT FOR TRE
 027D 65                OUT5                    ;THEN ECHO CHAR BACK
 027E 22                DEC     R2              ;UNDO AUTOINCR

                ;SHIFT RCVD 4 BIT VALUE INTO "IODAT" FROM R TO L

 027F 99                GHI     R9              ;GET BUFFER HB
 0280 FE                SHL
 0281 FE                SHL
 0282 FE                SHL
 0283 FE                SHL                     ;SHIFT L 4
 0284 52                STR     R2              ;PUT ON STACK
 0285 89                GLO     R9              ;BUFFER LB
 0286 F6                SHR
 0287 F6                SHR
 0288 F6                SHR
 0289 F6                SHR                     ;SHIFT R 4
 028A F1                OR                      ;MERGE HBLN+LBHN
 028B B9                PHI     R9              ;SAVE AS NEW HB

 028C 89                GLO     R9              ;BUFFER LB
 028D FE                SHL
 028E FE                SHL
 028F FE                SHL
 0290 FE                SHL                     ;SHIFT L 4
 0291 52                STR     R2              ;PUT ON STACK
 0292 8F                GLO     RF              ;RE-GET NEW NIBBLE
 0293 F1                OR                      ;MERGE LBHN+NEWNIB
 0294 A9                PLO     R9              ;SAVE AS NEW LB

 0295 305E              BR      ILOOP           ;GET NEXT DIGIT

 0000                   END
```

AVOCET SYSTEMS 1802 CROSS-ASSEMBLER - VERSION 1.40

TREC

---- SYMBOL TABLE ----

	002E	DECIML	0278	ILOOP	025E	NOTILT	00B1	PLAYBK	0008
	0000	DELOOP	0092	INDAT	025A	OCEXIT	00DD	PROMPT	015D
	002E	DLAY	0074	INEXIT	0259	OCLOOP	00E2	RESTRT	00A2
	0000	DLAYMK	0040	INIT	0131	ONEXIT	0045	RSLOOP	00A5
ACLOOP	0244	DTILT	009E	INLOOP	027B	ONLOOP	004E	SP	0020
AQLOOP	00C6	DYLOOP	007C	LF	000A	OTEXIT	002E	STKTOP	0100
AQUI	00B4	ENDCHK	00F6	LINSUM	020F	OUTCHR	00DE	TSEC	00E9
BURN	01AD	ETX	0003	MAIN	0054	OUTDAT	002F	TSEXIT	00E8
CAPGEE	0047	GDEXIT	00D4	MOVDAT	0014	OUTNUM	0046	TSLOOP	00EC
CKLOOP	0215	GETDAT	00D5	NOIDLE	006C	PARAM	0101	VERIFY	01B9
CR	000D	GLITCH	007B	NOINST	00C4	PBEND	0023	VLOOP	01C9

b. PLYBK

```
FL LOC   COSMAC CODE      LNNO SOURCE LINE
   0000                        ..PLAYBACK ROUTINE FOR DSDP TEMP RECORDER
   0000                        ..FOR STANDALONE USE
   0000                        ..IN EMERGENCY SITUATIONS
   0000                        ..I.E. WHEN "TREC" PLAYBACK ROUTINE FAILS.
   0000                        ..
   0000                        ..THIS PROGRAM CAN BE LOADED FROM
   0000                        ..THE SUPPORT COMPUTER AFTER INSTRUMENT
   0000                        ..RECOVERY.  IT DUMPS THE ENTIRE DATA AREA
   0000                        ..OF MEMORY IN THE USUAL FORMAT (SEE "TREC"
   0000                        ..PROGRAM SPEC).
   0000                        ..
   0000                        ..
   0000                        ..
   0000                        ..*****************   REGISTER USAGE   *********
   0000                        ..
   0000                        ..PROGRAM COUNTERS
   0000                        ..
   0000                        MAINPC = #00      ..MAINLINE ROUTINES PC
   0000                        TSECPC = #01      .."TSEC" PC
   0000                        OUTDPC = #04      .."OUTDAT" PC
   0000                        OUTCPC = #06      .."OUTCHR" PC
   0000                        OUTNPC = #03      .."OUTNUM"PC
   0000                        ..
   0000                        ..POINTERS AND FLAGS
   0000                        ..
   0000                        DATPTR = #07      ..DATA MEMORY POINTER
   0000                        STKPTR = #02      ..STACK POINTER
   0000                        ENDFLG = #0D      ..END-OF-DATA-AREA-FLAG
   0000                        ..
   0000                        ..BUFFERS AND WORKING STORAGE
   0000                        ..
   0000                        IODAT = #09       ..DATA FROM/TO I/O ROUTINES
   0000                        ..
   0000                        ..COUNTERS
   0000                        ..
   0000                        TSCTR = #0B       .."TSEC" INTERVAL COUNTER
   0000                        ..
   0000                        ..SCRATCHPAD
   0000                        ..
   0000                        SCRA   = #0F      ..SCRATCHPAD REGISTER
   0000                        ..
   0000                        ..
   0000                        ..
   0000                        ..*************   EQUATES AND CONSTANTS   *****
   0000                        ..
   0000                        ..I/O PORT NAMES
   0000                        ..
   0000                        UARTOT = #05      ..UART XMITTER BUFFER REGISTER
   0000                        POWRDN = #07      ..POWER DOWN
   0000                        ..
   0000                        ..CONSTANTS
   0000                        ..
   0000                        LF     = #0A      ..LINEFEED CHARACTER
   0000                        CR     = #0D      ..CARRIAGE RETURN
   0000                        SP     = #20      ..SPACE
   0000                        ETX    = #03      ..END-OF-TEXT CHARACTER
   0000                        ..
   0000                        ..
   0000                        ..*********************************************
   0000                        ..*****************   PROGRAM "PLYBK"   *******
   0000                        ..*********************************************
   0000                        ..
   0000                        ..
   0000                        ..INITIALIZE WORKING REGISTERS AS REQ'D
```

```
    0000                ..
F   0000 F800                       LDI A.1(TSEC)      ..INITIALIZE
    0002 B1                         PHI TSECPC         ..TSEC PC.
F   0003 F800                       LDI A.0(TSEC)
    0005 A1                         PLO TSECPC
    0006                ..
F   0006 F800                       LDI A.1(OUTDAT)    .."OUTDAT" PC
    0008 B4                         PHI OUTDPC
F   0009 F800                       LDI A.0(OUTDAT)
    000B A4                         PLO OUTDPC
    000C                ..
F   000C F800                       LDI A.1(OUTCHR)    .."OUTCHR" PC
    000E B6                         PHI OUTCPC
F   000F F800                       LDI A.0(OUTCHR)
    0011 A6                         PLO OUTCPC
    0012                ..
F   0012 F800                       LDI A.1(OUTNUM)
    0014 B3                         PHI OUTNPC
F   0015 F800                       LDI A.0(OUTNUM)
    0017 A3                         PLO OUTNPC
    0018                ..
F   0018 F800                       LDI A.1(STKTOP)    ..STACK POINTER
    001A B2                         PHI STKPTR
F   001B F800                       LDI A.0(STKTOP)
    001D A2                         PLO STKPTR
    001E                ..
    001E E2                          SEX STKPTR
    001F                ..
    001F F80B                        LDI #0B          ..END-OF-DATA FLAG
    0021 BD                          PHI ENDFLG
    0022 F8FF                        LDI #FF
    0024 AD                          PLO ENDFLG
    0025                ..
F   0025 F800                        LDI A.1(PARAM)        ..DATA POINTER
    0027 B7                          PHI DATPTR
F   0028 F800                        LDI A.0(PARAM)
    002A A7                          PLO DATPTR
    002B                ..
    002B                ..COMPARE "DATPTR" TO "ENDFLG" TO CHECK FOR END-
    002B                ..
    002B 97             PLAYBK:     GHI DATPTR         ..GET DATA PTR
    002C 52                         STR STKPTR         ..PUT ON STACK, NO PU
    002D 9D                         GHI ENDFLG         ..GET FLAG
    002E F7                         SM               ..SUBTRACT FLG - PTR
F   002F 3A00                       BNZ MOVDAT         ..IF .NE. DO DATA OUT
    0031                ..
    0031 87                         GLO DATPTR         ..ELSE CHECK
    0032 52                         STR STKPTR         ..LOW BYTES ALSO
    0033 8D                         GLO ENDFLG
    0034 F7                         SM                 ..IF LB'S MATCH,
F   0035 3200                       BZ PBEND           ..GO END PLAYBACK.
    0037                                               ..ELSE NEXT DATA PT.
    0037                ..
    0037                ..MOVE DATA TO "IODAT",SEND WITH "OUTDAT", SEND
    0037                ..
    0037 47             MOVDAT:     LDA DATPTR         ..MOVE 2 BYTES
    0038 B9                         PHI IODAT           ..(1 SAMPLE) INTO
    0039 47                         LDA DATPTR          ..I/O BUFFER
    003A A9                         PLO IODAT           ..AND SEND WITH
    003B D4                         SEP OUTDPC          ..CALL "OUTDAT"
    003C                ..
    003C D6                         SEP OUTCPC           ..CALL "OUTCHR" TO
    003D 0D0A00                     ,CR,LF,#00.          ..SEND CR,LF
    0040                ..
    0040 F800                        LDI #00
    0042 BB                          PHI TSCTR
```

```
0043 F802                   LDI #02              ..DELAY 0.2 SEC
0045 AB                     PLO TSCTR            ..BY CALL TO
0046 D1                     SEP TSECPC           .."TSEC"
0047                ..
0047 302B                   BR PLAYBK            ..KEEP SENDING DATA
0049                ..
0049 D6             PBEND:  SEP OUTCPC           ..SEND MSG. WHEN THR
004A 53544F50               ,T'STOP'
004E 030D0A00               ,ETX,CR,LF,#00
0052                ..
0052 6F                     INP POWRDN           ..THEN POWER DOWN
0053 00                     IDLE                 ..AND IDLE FOR SAFET
0054                ..
0054                ..
0054                ..******************   OUTDAT/OUTNUM   *******
0054                ..
0054                .."OUTDAT" TRANSMITS CONTENTS OF REGISTER "IODAT
0054                .."OUTDAT" ENTERED VIA "SEP OUTDPC" AND EXITS VI
0054                ..
0054 D0             OTEXIT:  SEP MAINPC          ..RETURN TO CALLER
0055                ..
0055 99             OUTDAT:  GHI IODAT           ..HIGH BYTE TO XMIT
0056 F6F6F6F6                SHR;SHR;SHR;SHR     ..ISOLATE HIGH NIBBL
005A D3                      SEP OUTNPC          ..SEND IT
005B                ..
005B 99                      GHI IODAT           ..HIGH BYTE
005C FA0F                    ANI #0F             ..LOW NIBBLE
005E D3                      SEP OUTNPC
005F                ..
005F 89                      GLO IODAT           ..LOW BYTE
0060 F6F6F6F6                SHR;SHR;SHR;SHR     ..HIGH NIBBLE
0064 D3                      SEP OUTNPC
0065                ..
0065 89                      GLO IODAT           ..LOW BYTE
0066 FA0F                    ANI #0F             ..LOW NIBBLE
0068 D3                      SEP OUTNPC
0069                ..
0069 3054                    BR OTEXIT           ..RESET "OUTDPC" PER
006B                ..
006B                .."OUTNUM" -- DEDICATED SUBROUTINE FOR "OUTDAT".
006B                ..VALUE IN D REG TO ASCII HEX (0-9, A-F) AND XMI
006B                ..
006B D4             ONEXIT:  SEP OUTDPC          ..RETN TO "OUTDAT"
006C                ..
006C FF0A           OUTNUM:  SMI #0A             ..TEST FOR <= 9
006E C7                      LSNF                ..SKIP ADD IF YES
006F FC07                    ADI #07             ..IF A-F, ADD EXTRA
0071                ..
0071 FC3A                    ADI #3A             ..ADD NET 30 OR
0073                                             ..NET 36 TO GET
0073                .                            ..ASCII CHARACTER.
0073 52                      STR STKPTR          ..PUT ON STACK, NO P
0074 3474           ONLOOP:  B1 ONLOOP           ..WAIT FOR UART TPE
0076 65                      OUT UARTOT          ..SEND TO UART
0077 22                      DEC STKPTR          ..UNDO AUTOINCREMENT
0078 306B                    BR ONEXIT           ..RESET "OUTNPC".
007A                ..
007A                ..*******************   OUTCHR   **************
007A                ..
007A                ..      ENTERED VIA "SEP OUTCPC".  OUTPUTS CHARAC
007A                ..POINTED TO BY "MAINPC" UNTIL A 00 BYTE IS ENCO
007A                ..RETURNS VIA "SEP MAINPC".  ALTERS M(STKPTR)
007A                ..
007A D0             OCEXIT:  SEP MAINPC          ..RETURN TO CALLER
007B 40             OUTCHR:  LDA MAINPC          ..GET CHARACTER
007C 327A                    BZ  OCEXIT          ..IF #00, EXIT
```

```
007E 52                       STR STKPTR      ..ELSE PUT ON STACK
007F 347F       OCLOOP:       B1  OCLOOP      ..WAIT FOR UART
0081 65                       OUT UARTOT      ..XMIT WHEN READY
0082 22                       DEC STKPTR      ..UNDO AUTOINCR.
0083 307B                     BR OUTCHR       ..GET NEXT CHAR
0085            ..
0085            ..***************   TSEC   *******************
0085            ..
0085            .."TSEC" IS A 0.1 SEC TIMING ROUTINE.  TIME FROM
0085            ..EXECUTION OF FIRST INSTRUCTION (DEC TSCTR) TO
0085            ..OF LAST INSTRUCTION (SEP MAINPC) IS
0085            ..      (0.1) X (VALUE IN "TSCTR") SECONDS,
0085            ..ASSUMING A 38.400 KHZ CPU CLOCK.
0085            ..CALL WITH "SEP TSECPC".  RETURNS TO MAINPC =0
0085            ..
0085 D0         TSEXIT:       SEP MAINPC      ..RETURN TO CALLER
0086 2B         TSEC:         DEC TSCTR       ..DEC COUNTER
0087 F874                     LDI #74         ..SET UP LOCAL LOOP
0089            ..
0089 FF01       TSLOOP:       SMI #01
008B 3A89                     BNZ TSLOOP
008D            ..
008D 9B                       GHI TSCTR       ..CHECK COUNTER
008E 52                       STR STKPTR      ..PUT HB ON STACK
008F 8B                       GLO TSCTR       ..GET LB .
0090 F1                       OR              ..MERGE HB+LB
0091 3285                     BZ TSEXIT       ..EXIT IF BOTH 0
0093            ..
0093 3086                     BR TSEC         ..ELSE KEEP LOOPING
0095            ..********************   DATA AREA   ***********
0095            ..
0100                          ORG #100
0100            STKTOP:
0101                          ORG *+1
0101            PARAM:
0101            END
NO UNDEFINED NAMES
```

c. ADTEST

```
ADDRESS   CODE MNEMONIC COMMENT

                        ;NAME OF PROGRAM, ADTEST
                        ;*************************************
0000      F8   LDI      ;MAKE REG 3 = 00C8
0001      C8   C8       ;MAKE REG THE ADDRESS OF X OUT IN THE BOONIES
                        ;WHERE IT WON'T HURT ANYTHING
0002      A3   PLO      ;PUT LOW REG 3
0003      F8   LDI
0004      00   00
0005      B3   PHI      ;PUT HI REG 3
0006      E3   SEX      ;TO REG 3
0007      7B   SET Q    ;TURN ON A/D POWER
                        ;**************************************
                        ;START A/D CYCLE AGAIN
0008      6B   INP 3    ;TRIGGER A/D
                        ;WAIT LOOP FOR MEASUREMENT REPETITION
                        ;RATE
0009      F8   LDI
000A      FF   FF       ;WAIT TIME
000B      FF   SMI      ;SUBTRACT MEMORY IMMEDIATE
000C      01   01       ;NUMBER SUBTRACTED
000D      C4   NOP      ;WASTE TIME
000E      C4   NOP
000F      C4   NOP
0010      C4   NOP
0011      C4   NOP
0012      C4   NOP
0013      C4   NOP
0014      C4   NOP
0015      3A   BNZ      ;SHORT BRANCH IF NOT 0
0016      0B   0B       ;BRANCH LOCATION
                        ;*********************************
                        ;READ A/D AND SEND IT OUT VIA UART
0017      3E   BN3      ;SHORT BRANCH IF EF3=0, WAIT IF A/D STATUS IS
                        ;HI
0018      17   17       ;BRANCH LOCATION
0019      69   INP 1    ;INPUT 1, HI ORDER BYTE FROM A/D
001A      65   OUT 5    ;OUTPUT 5, OUTPUT HI ORDER BYTE TO UART
001B      23   DEC      ;DECREMENT REG 3, CANCELS AUTO INCREMENT IN
                        ;REG 3 DUE TO OUT 5
001C      34   B1       ;SHORT BRANCH IF EF1=1, WAIT TIL UART READY
001D      1C   1C       ;BRANCH LOCATION
001E      6A   INP 2    ;INPUT 2, LO ORDER A/D BYTE
001F      65   OUT 5    ;OUTPUT 5, OUTPUT LO ORDER BYTE TO UART
0020      23   DEC      ;DECREMENT REG 3 (X REG), CANCELS AUTO
                        ;INCREMENT DUE TO OUT 5
0021      34   B1       ;SHORT BRANCH IF EF1=1, WAIT FOR UART TO
                        ;FINISH
0022      21   21       ;BRANCH LOCATION
                        ;***********************************
                        ;END OF READING OUT A/D
0023      30   BR       ;SHORT BRANCH, START AGAIN
0024      08   08       ;BRANCH LOCATION
```

d. MEMTEST2

```
ADDR    CODE    LABEL   MNEMONIC     COMMENT
                                     ;PROGRAM NAME - MEMTST2
0000    C4              NOP
0001    C4              NOP
0002    7B              SEQ           ;TURN OFF A/D CONV
0003    F8AA            LDI #AA      ;LOAD IN TEST PATTERN
0005    A4              PLO R4       ;SAVE IT IN R4.0
                                     ;*******************************
                                     ;START MEMORY TEST
0006    F800    START   LDI #00
0008    B3              PHI R3       ;SET UP OUTPUT POINTER
0009    A3              PLO R3
000A    B2              PHI R2       ;SET UP MEMORY TEST POINTER
000B    F860            LDE #60
000D    A2              PLO R2
                                     ;**********************************
                                     ;FILL MEMORY WITH TEST PATTERN
000E    84      FILL    GLO R4       ;LOAD TEST WORD
000F    52              STR R2       ;PUT IN MEMORY
0010    12              INC R2       ;POINT UP TO NEXT LOCATION
0011    92              GHI R2       ;GET MEMORY PTS
0012    FF0C            SMI #0C      ;TEST FOR DONE
0014    3B0E            BM  FILL     ;IF NO, KEEP FILLING
                                     ;***********************************
0016    F800            LDI #00      ;RESET R2 TO BEGINNING OF TEST AREA
0018    B2              PHI R2
0019    F860            LDI #60
001B    A2              PLO R2
                                     ;**********************************
                                     ;COMPARE MEMORY WITH TEST PATTERN
001C    E2              SEX R2       ;SET R2=X
001D    84      TEST    GLO R4       ;GET TEST PATTERN FROM R4.0
001E    F3              XOR          ;COMPARE TEST PATN & MEM CONTENT
001F    3A35            BNZ NOTOK    ;IF "NOT OK," BRANCH
0021    12      INCR    INC R2       ;ELSE INCREMENT MEM POINTER
0022    92              GHI R2       ;GET POINTER
0023    FF0C            SMI #0C      ;TEST FOR DONE
0025    3B1D            BM TEST      ;KEEP TESTING IF NOT DONE
0027    F847            LDI #47      ;LOAD "G"
0029    53              STR R3
002A    E3              SEX R3
002B    342B    GLOOP   B1  GLOOP    ;WAIT FOR UART FINISH
002D    65              OUT 5        ;OUTPUT TO UART
002E    23              DEC R3       ;UNDO AUTO-INCREMENT
002F    84              GLO R4       ;GET TEST PATTERN
0030    FBFF            XRI #FF      ;COMPLEMENT IT
0032    A4              PLO R4       ;STORE NEW TEST BYTE
0033    3006            BR  START    ;REPEAT TEST
                                     ;*******************************
                                     ;MEMORY ADDRESS AND ERROR OUTPUT
                                     ;ROUTINE
                                     ;
0035    E3      NOTOK   SEX R3       ;POINT TO OUTPUT REG
0036    53              STR R3       ;STORE ERROR WORD IN OUTPUT REG
0037    3437    BLOOP   B1  BLOOP    ;WAIT UNTIL UART READY
0039    65              OUT 5        ;UART OUTPUT
```

```
003A    23                DEC R3      ;UNDO AUTO INCREMENT
                                      ;******************************
003B    92                GHI R2      ;UPPER BYTE OF MEM TEST LOCATION
003C    53                STR R3      ;OUTPUT TO UART
003D    343D    CLOOP     B1  CLOOP
003F    65                OUT 5
0040    23                DEC R3
                                      ;******************************
0041    82                GLO R2      ;LOWER BYTE OF MEM TEST LOCATION
0042    53                STR R3      ;OUTPUT TO UART
0043    3443    DLOOP     B1  DLOOP
0045    65                OUT 5
0046    23                DEC R3
                                      ;******************************
0047    F800              LDI #00     ;DELIMITER CHARACTER
0049    53                STR R3      ;OUTPUT TO UART
004A    344A    ELOOP     B1  ELOOP
004C    65                OUT 5
004D    23                DEC R3
                                      ;*******************************
004E    E2                SEX R2      ;RESET R2=2 FOR TEST
004F    3021              BR  INC     ;RETURN TO TEST ROUTINE
0051    C4                NOP
```

3. MACHINE CODE FOR TEMPERATURE RECORDER (xxx.ASC FILES)

a. TREC.ASC

```
TREC.ASC PROGRAM
0000:C0 01 31 E2 98 B7 88 A7
0008:97 52 9D F7 3A 14 87 52
0010:8D F7 32 23 47 B9 47 A9
0018:D4 D6 0D 0A 00 F8 02 AB
0020:D1 30 08 D6 53 54 4F 50
0028:03 0D 0A 00 6F 00 D0 99
0030:F6 F6 F6 F6 D3 99 FA 0F
0038:D3 89 F6 F6 F6 F6 D3 89
0040:FA 0F D3 30 2E D4 FF 0A
0048:C7 FC 07 FC 3A 52 34 4E
0050:65 22 30 45 48 BD 52 48
0058:AD F1 3A 6C BF AF F8 C4
0060:5F 1F 5F 1F 5F 97 BD 87
0068:AD 7A 6F 00 48 B9 48 A9
0070:48 BA 48 AA 9A BC 8A AC
0078:F8 3C AE E2 9C 52 8C F1
0080:32 84 8E 32 A2 6E F8 40
0088:52 65 22 2C F8 09 AB D1
0090:F8 49 FF 01 7A 3A 92 3F
0098:9E F8 3C AE 30 7C 2E 7A
00A0:30 7C F8 4B AE 8E 32 74
00A8:F8 28 AB D1 37 B1 F8 4C
00B0:AE 2E 30 A5 9D 52 8D F1
00B8:32 54 7B 6E 2D 99 BB 89
00C0:AB 2B D1 6B F8 6A FF 01
00C8:3A C6 D5 97 FF 0C 3B B4
00D0:27 27 30 54 D0 69 57 17
00D8:6A 57 17 30 D4 D0 40 32
00E0:DD 52 34 E2 65 22 30 DE
00E8:D0 2B F8 74 FF 01 3A EC
00F0:9B 52 88 F1 32 E8 30 E9
00F8:FF FF FF FF FF FF FF FF
0100:FF FF FF FF FF FF FF FF
0108:FF FF FF FF FF FF FF FF
0110:FF FF FF FF FF FF FF FF
0118:FF FF FF FF FF FF FF FF
0120:FF FF FF FF FF FF FF FF
0128:FF FF FF FF FF FF FF FF
0130:FF 7A F8 00 B1 F8 E9 A1
0138:F8 02 B3 F8 5A A3 F8 00
0140:B4 F8 2F A4 F8 00 B6 F8
0148:DE A6 F8 00 B5 F8 D5 A5
0150:F8 01 B2 F8 00 A2 F8 01
0158:B8 F8 01 A8 E2 D6 0A 0D
0160:00 F8 00 BB F8 02 AB D1
0168:D6 23 53 41 4D 50 4C 45
0170:53 3F 20 03 00 D3 99 58
0178:52 18 89 58 18 F1 32 AD
0180:D6 20 20 20 49 4E 54 45
0188:52 56 41 4C 3F 20 03 00
0190:D3 99 58 18 89 58 18 D6
0198:20 20 20 44 45 4C 41 59
01A0:3F 20 03 00 D3 99 58 18
01A8:89 58 18 30 5D F8 00 BF
01B0:AF 1F F8 00 5F 1F F8 7B
01B8:5F F8 00 B3 F8 46 A3 F8
01C0:01 BB F8 01 A8 D6 0A 0A
01C8:00 D6 0A 0D 00 F8 02 AB
01D0:D1 48 B9 52 48 A9 F1 C2
01D8:02 0F D6 23 53 41 4D 50
01E0:4C 45 53 3D 20 00 D4 48
01E8:B9 48 A9 D6 20 20 20 49
01F0:4E 54 45 52 56 41 4C 3D
01F8:20 00 D4 48 B9 48 A9 D6
0200:20 20 20 44 45 4C 41 59
0208:3D 20 00 D4 C0 01 C9 F8
0210:00 B7 A7 B9 E7 89 F4 A9
0218:17 97 FF 00 CB 02 15 87
0220:FF F6 CB 02 15 E2 D6 20
0228:20 20 43 48 45 43 4B 53
0230:55 4D 3D 20 00 D4 D6 20
0238:20 20 41 43 54 49 4F 4E
0240:3F 20 03 00 35 44 6C FF
0248:47 CA 01 31 98 B7 88 A7
0250:F8 01 B8 F8 01 A8 C0 00
0258:54 D0 F8 00 B9 A9 35 5E
0260:6C FF 0D 32 59 F0 FF 30
0268:3B 5E FF 0A 3B 78 FF 07
0270:3B 5E FF 06 33 5E FC 06
0278:FC 0A AF 34 7B 65 22 99
0280:FE FE FE FE 52 89 F6 F6
0288:F6 F6 F1 B9 89 FE FE FE
0290:FE 52 3F F1 A9 30 5E FF
0298:FF FF FF FF FF FF FF ZZ
```

b. AVERG.ASC

```
AVERG.ASC PROGRAM
0000:C0 01 A2 E2 98 B7 88 A7
0008:97 52 9D F7 3A 14 87 52
0010:8D F7 32 23 47 B9 47 A9
0018:D4 D6 0D 0A 00 F8 02 AB
0020:D1 30 08 D6 53 54 4F 50
0028:03 0D 0A 00 6F 00 D0 99
0030:F6 F6 F6 F6 D3 99 FA 0F
0038:D3 89 F6 F6 F6 F6 D3 89
0040:FA 0F D3 30 2E D4 FF 0A
0048:C7 FC 07 FC 3A 52 34 4E
0050:65 22 30 45 48 BD 52 48
0058:AD F1 3A 6C BF AF F8 C4
0060:5F 1F 5F 1F 5F 97 BD 87
0068:AD 7A 6F 00 48 B9 48 A9
0070:48 8A 48 AA 9A BC 8A AC
0078:F8 3C AE E2 9C 52 8C F1
0080:32 B4 8E 32 A2 6E F8 40
0088:52 65 22 2C F8 09 AB D1
0090:F8 49 FF 01 7A 3A 92 3F
0098:9E F8 3C AE 30 7C 2E 7A
00A0:30 7C F8 4B AE 8E 32 74
00A8:F8 28 AB D1 37 B1 F8 4C
00B0:AE 2E 30 A5 99 BB 89 AB
00B8:7B 7B 9D 52 8D F1 32 54
00C0:F8 00 BA AA F8 53 52 65
00C8:22 D5 2B 9B 52 8B F1 32
00D0:D9 F8 15 FF 01 3A D3 30
00D8:B9 9A FA 80 FB 80 73 3A
00E0:EC 2A 9A FB FF BA 8A FB
00E8:FF AA 30 F2 C4 C4 C4 C4
00F0:7B 7B 12 9A F6 F6 F6 F1
00F8:57 17 8A F6 F6 F6 52 9A
0100:FE FE FE FE FE F1 57 17
0108:2D 97 FF 0C CB 00 B4 27
0110:27 C0 00 54 D0 F8 08 AC
0118:8C 32 14 2C 6B 7B F8 86
0120:FF 01 3A 20 69 FA 80 32
0128:3A 6A 8A F4 AA 69 FA 0F
0130:52 9A 74 BA 7B 7B 7B 7B
0138:30 18 6A FB FF FC 01 52
0140:8A 74 AA 69 FA 0F FB FF
0148:52 9A 74 BA 30 18 D0 40
0150:32 4E 52 34 53 65 22 30
0158:4F D0 2B F8 74 FF 01 3A
0160:5D 9B 52 8B F1 32 59 30
0168:5A FF FF FF FF FF FF FF
0170:FF FF FF FF FF FF FF FF
0178:FF FF FF FF FF FF FF FF
0180:FF FF FF FF FF FF FF FF
0188:FF FF FF FF FF FF FF FF
0190:FF FF FF FF FF FF FF FF
0198:FF FF FF FF FF FF FF FF
01A0:FF FF 7A F8 01 B1 F8 5A
01A8:A1 F8 02 B3 F8 CD A3 F8
01B0:00 B4 F8 2F A4 F8 01 B6
01B8:F8 4F A6 F8 01 B5 F8 15
01C0:A5 F8 01 B2 F8 71 A2 F8
01C8:01 B8 F8 72 A8 E2 D6 0A
01D0:0D 00 F8 00 BB F8 02 AB
01D8:D1 D6 23 53 41 4D 50 4C
01E0:45 53 3F 20 03 00 D3 99
01E8:58 52 18 89 58 18 F1 C2
01F0:02 20 D6 20 20 20 49 4E
01F8:54 45 52 56 41 4C 3F 20
0200:03 00 D3 99 58 18 89 58
0208:18 D6 20 20 20 44 45 4C
0210:41 59 3F 20 03 00 D3 99
0218:58 18 89 58 18 C0 01 CE
0220:F8 00 BF AF 1F F8 00 5F
0228:1F F8 7B 5F F8 00 B3 F8
0230:46 A3 F8 01 B8 F8 72 A8
0238:D6 0A 0A 00 D6 0A 0D 00
0240:F8 02 AB D1 48 B9 52 48
0248:A9 F1 C2 02 82 D6 23 53
0250:41 4D 50 4C 45 53 3D 20
0258:00 D4 48 B9 48 A9 D6 20
0260:20 20 49 4E 54 45 52 56
0268:41 4C 3D 20 00 D4 48 B9
0270:48 A9 D6 20 20 20 44 45
0278:4C 41 59 3D 20 00 D4 C0
0280:02 3C F8 00 B7 A7 B9 E7
0288:89 F4 A9 17 97 FF 01 CB
0290:02 88 87 FF 67 CB 02 88
0298:E2 D6 20 20 20 43 48 45
02A0:43 4B 53 55 4D 3D 20 00
02A8:D4 D6 20 20 20 41 43 54
02B0:49 4F 4E 3F 20 03 00 35
02B8:B7 6C FF 47 CA 01 A2 98
02C0:B7 88 A7 F8 01 B8 F8 72
02C8:A8 C0 00 54 D0 F8 00 B9
02D0:A9 35 D1 6C FF 0D 32 CC
02D8:F0 FF 30 3B D1 FF 0A 3B
02E0:EB FF 07 3B D1 FF 06 33
02E8:D1 FC 06 FC 0A AF 34 EE
02F0:65 22 99 FE FE FE FE 52
02F8:89 F6 F6 F6 F6 F1 B9 89
0300:FE FE FE FE 52 8F F1 A9
0308:C0 02 D1 FF FF FF FF FF
0310:00 ZZ
```

c. PLYBK.ASC

```
PLYBK.ASC PROGRAM
0000:F8 00 B1 F8 86 A1 F8 00
0008:B4 F8 55 A4 F8 00 B6 F8
0010:7B A6 F8 00 B3 F8 6C A3
0018:F8 01 B2 F8 00 A2 E2 F8
0020:0B BD F8 FF AD F8 01 B7
0028:F8 01 A7 97 52 9D F7 3A
0030:37 87 52 8D F7 32 49 47
0038:B9 47 A9 D4 D6 0D 0A 00
0040:F8 00 BB F8 02 AB D1 30
0048:2B D6 53 54 4F 50 03 0D
0050:0A 00 6F 00 D0 99 F6 F6
0058:F6 F6 D3 99 FA 0F D3 89
0060:F6 F6 F6 F6 D3 89 FA 0F
0068:D3 30 54 D4 FF 0A C7 FC
0070:07 FC 3A 52 34 74 65 22
0078:30 6B D0 40 32 7A 52 34
0080:7F 65 22 30 7B D0 2B F8
0088:74 FF 01 3A 89 9B 52 8B
0090:F1 32 85 30 86 FF FF FF
0098:FF FF FF FF FF FF FF FF
00A0:ZZ
```

d. ADTEST.ASC

```
ADTEST.ASC PROGRAM
0000:F8 C8 A3 F8 00 B3 E3 7B
0008:6B F8 A8 FF 01 C4 C4 C4
0010:C4 C4 C4 C4 C4 3A 0B 3E
0018:17 69 65 23 34 1C 6A 65
0020:23 34 21 30 08 ZZ
```

e. MEMTS2.ASC

```
MEMTST2.ASC PROGRAM
0000:C4 C4 7B F8 AA A4 F8 00
0008:B3 A3 B2 F8 60 A2 84 52
0010:12 92 FF 0C 3B 0E F8 00
0018:B2 F8 60 A2 E2 84 F3 3A
0020:35 12 92 FF 0C 3B 1D F8
0028:47 53 E3 34 2B 65 23 84
0030:FB FF A4 30 06 E3 53 34
0038:37 65 23 92 53 34 3D 65
0040:23 82 53 34 43 65 23 F8
0048:00 53 34 4A 65 23 E2 30
0050:21 C4 ZZ
```

APPENDIX G. PRO-350 PROGRAMS

1. PROBASIC.BAS

```
SET NO DOUBLE
5 PROGRAM PROBASIC
10 PRINT "FOR HEATFLOW TYPE 'RUN MENU' <CR>"
```

2. MENU.BAS

```
SET NO DOUBLE
5 PROGRAM MENU
10 HOME$=CHR$(27)+"[2J"+CHR$(27)+"[0;0H"\INVERSE$=CHR$(27)+"[7m"
15 FLASH$=CHR$(27)+"[5m"\NORMAL$=CHR$(27)+"[0m"\CENTER$=CHR$(27)+"[12;30H"
20 PRINT HOME$\ PRINT TAB(38);INVERSE$;"MENU";NORMAL$\ PRINT \ PRINT
30 PRINT "1.  THERMAL CONDUCTIVITY OPERATIONS"
40 PRINT "2.  DOWN HOLE INSTRUMENT OPERATIONS"
99 PRINT "9.  EXIT BASIC"
100 PRINT \ PRINT FLASH$;"INPUT YOUR SELECTION ";NORMAL$;\ INPUT A$
110 IF A$<"1" OR A$>"9" THEN GOTO 20 ELSE A=VAL(A$)
120 ON A GOTO 140,150,20,20,20,20,20,20,220
130 GOTO 20
140 PRINT HOME$;CENTER$;INVERSE$;"GETTING THERMCON PROGRAM";NORMAL$
145 CHAIN "THERMCON.BAS"
150 PRINT HOME$;CENTER$;INVERSE$;"GETTING LOADER PROGRAM";NORMAL$
155 CHAIN "LOADER.BAS"
220 END
```

3. LOADER.BAS

```
SET NO DOUBLE
5  PROGRAM LOADER
10  REM  DSDP INSTRUMENT PROGRAM MODIFIED FOR PRO-350
11  REM BY GEORGE L. PELLETIER WHOI 19 DEC 85
20  HOME$=CHR$(27)+"[2J"+CHR$(27)+"[0;0H"\INVERSE$=CHR$(27)+"[7m"
25  NORMAL$=CHR$(27)+"[0m"\FLASH$=CHR$(27)+"[5m"\CENTER$=CHR$(27)+"[12;30H"
30  PRINT HOME$+INVERSE$+"* CORING TOOL TEMPERATURE RECORDER *"+NORMAL$\ PRINT
35  PRINT "MENU"\ PRINT "1. LOAD PROGRAM"\ PRINT "2. LOAD PARAMETERS"
40  PRINT "3. DUMP DATA"\ PRINT "4. DISPLAY FILE DATA"
45  PRINT "5. PRINT FILE DATA"
50  PRINT "6. CONVERT & PRINT FILE DATA"\ PRINT "7. END"
100  PRINT FLASH$;"ENTER YOUR CHOICE ";NORMAL$;\ INPUT A$\A=VAL(A$)
110  ON A GOTO 120,1000,2000,3000,4000,5000,6000
115  GOTO 30
120  PRINT HOME$+INVERSE$+"*** LOAD PROGRAM ***"+NORMAL$\ PRINT
125  PRINT "1. LOAD MEMORY TEST"\ PRINT "2. LOAD MAIN PROGRAM"
130  PRINT "3. LOAD A/D TEST"\ PRINT "4. LOAD AVERAGING PROGRAM"
135  PRINT "5. LOAD PLAYBACK PROGRAM"
150  PRINT FLASH$;"ENTER YOUR CHOICE ";NORMAL$;\ INPUT A$\A=VAL(A$)
155  IF A<1 OR A>5 THEN 30
160  IF A$="1" THEN PG$="MEMTS2.ASC"\ GOTO 250
170  IF A$="2" THEN PG$="TREC2.ASC"\ GOTO 250
180  IF A$="3" THEN PG$="ADTEST.ASC"\ GOTO 250
190  IF A$="4" THEN PG$="AVERG.ASC"\ GOTO 250
200  IF A$="5" THEN PG$="PLYBK.ASC"\ GOTO 250
250  OPEN PG$ FOR INPUT AS FILE #1%\ REM GET READY TO LOAD PROGRAM
255  PRINT \ PRINT INVERSE$;"SWITCH INSTRUMENT FROM RUN TO LOAD THEN PRESS RETURN";NORMAL$
256  INPUT A$
260  PRINT HOME$+CENTER$+INVERSE$+"LOADING "+PG$+NORMAL$
270  GOSUB 10000\ GOTO 30
1000  CHAIN "LOAD1.BAS" WITH HOME$,INVERSE$,NORMAL$,FLASH$,CENTER$
2000  CHAIN "LOAD2.BAS" WITH HOME$,INVERSE$,NORMAL$,FLASH$,CENTER$
3000  CHAIN "LOAD3.BAS" WITH HOME$,INVERSE$,NORMAL$,FLASH$,CENTER$
4000  CHAIN "LOAD3A.BAS" WITH HOME$,INVERSE$,NORMAL$,FLASH$,CENTER$
5000  CHAIN "LOAD4.BAS" WITH HOME$,INVERSE$,NORMAL$,FLASH$,CENTER$
6000  PRINT HOME$;CENTER$;INVERSE$;"RETURNING TO MENU";NORMAL$
6005  CHAIN "MENU.BAS"
6010  END
10000  REM LOAD PROGRAM INTO DOWN HOLE INSTRUMENT SBR.
10100  INPUT #1%,A$
10150  OPEN "XK:" AS FILE #2%\ REM COMMUNICATIONS PORT
10200  FOR N=6 TO LEN(A$) STEP 3
10300  VAR$=MID$(A$,N,2)\ IF VAR$="ZZ" THEN CLOSE #1%\ GOTO 11500
10500  FOR I1=2 TO 1 STEP -1\NUM$=MID$(VAR$,I1,1)
10510  IF NUM$<="9" THEN NUMBER=VAL(NUM$) ELSE NUMBER=ASCII(NUM$)-55
10515  IF I1=2 THEN TOTAL=NUMBER ELSE IF I1=1 THEN TOTAL=TOTAL+(NUMBER*16)
10550  NEXT I1
10600  PRINT #2%,CHR$(TOTAL);
11460  NEXT N
11470  GOTO 10100
11500  CLOSE #1%,#2%
11510  PRINT HOME$+CHR$(27)+"[12B"+CHR$(27)+"[12C"+FLASH$;
11520  PRINT "PROGRAM TRANSMITTED"+NORMAL$\ PRINT
11530  PRINT INVERSE$;"SWITCH INSTRUMENT TO RESET";NORMAL$
11550  PRINT "PRESS RETURN TO CONTINUE ";\ INPUT A$
11600  RETURN
```

4. LOAD1.BAS

```
SET NO DOUBLE
1000 PROGRAM LOAD1(HOME$,INVERSE$,NORMAL$,FLASH$,CENTER$)
1001 REM 19 DEC 85
1005  PRINT HOME$+INVERSE$+"*** LOAD PARAMETERS ***"+NORMAL$
1010 OPEN "XK:" AS FILE #2%\ REM COMMUNICATIONS PORT
1015 PRINT \ PRINT INVERSE$+"MOVE INTERFACE SWITCH FROM RESET TO RUN";NORMAL$
1017 LINPUT #2%,A$
1018 N2=0
1019 B$=""
1020 LINPUT #2%,A$\ IF LEN(A$)<1 THEN 1020 ELSE B$=B$+A$
1025 IF ASCII(MID$(A$,LEN(A$),1))=3 THEN GOSUB 1200\ GOTO 1027 ELSE 1020
1027 PRINT B$;"(DECIMAL) ";\N2=N2+1\ IF N2=4 THEN N2=1
1030 INPUT A$\DEC=VAL(A$)\ GOSUB 1500\ FOR I1=1 TO LEN(HEX$)
1031 PRINT #2%,MID$(HEX$,I1,1)\ GOSUB 1033\ NEXT I1\ GOSUB 1033
1032 PRINT #2%,CHR$(13)\ GOTO 1040
1033 FOR I2=1 TO 50\ NEXT I2\ RETURN
1040 IF HEX$="0000" AND N2=1 THEN 1045 ELSE 1019
1045 N=1
1049 B$=""
1050 LINPUT #2%,A$\ IF LEN(A$)<1 THEN 1050 ELSE B$=B$+A$
1060 IF ASCII(MID$(A$,LEN(A$),1))=13 THEN C$(N)=B$\N=N+1\B$=""
1070 IF ASCII(MID$(A$,LEN(A$),1))=3 THEN C$(N)=B$\ GOTO 1090
1080 GOTO 1050
1090 FOR N1=2 TO N-1
1100 SAM$=MID$(C$(N1),11,4)\INTVL$=MID$(C$(N1),28,4)\DELY$=MID$(C$(N1),42,4)
1110 HEX$=SAM$\ GOSUB 1600\ PRINT MID$(C$(N1),1,10)+DEC$;
1120 HEX$=INTVL$\ GOSUB 1600\ PRINT MID$(C$(N1),15,13)+DEC$;
1130 HEX$=DELY$\ GOSUB 1600\ PRINT MID$(C$(N1),32,10)+DEC$
1140 NEXT N1\ PRINT C$(N)
1150 CALL INKEY (A$)\ IF LEN(A$)=0 THEN 1150
1160 IF ASCII(A$)=13 THEN PRINT #2%,CHR$(13)\ GOTO 1017
1170 IF A$="G" THEN PRINT #2%,"G"
1180 PRINT HOME$+CENTER$+INVERSE$+"INSTRUMENT RUNNING"+NORMAL$\ PRINT
1190 PRINT "PRESS RETURN FOR MENU";\ INPUT A$
1195 CHAIN "LOADER.BAS"
1200 REM B$ CLEANUP
1210 IF LEN(B$)>11 THEN B$=MID$(B$,LEN(B$)-10,11)
1220 RETURN
1500 REM DEC TO HEX CONVERT
1505 HEX$=""
1510 FOR N=12 TO 4 STEP -4
1515 HEX=0
1520 IF DEC<2^N THEN 1530 ELSE DEC=DEC-2^N\HEX=HEX+1\ GOTO 1520
1530 IF HEX<10 THEN HEX$=HEX$+CHR$(HEX+48)
1540 IF HEX>9 THEN HEX$=HEX$+CHR$(HEX+55)
1550 NEXT N
1560 IF DEC<10 THEN HEX$=HEX$+CHR$(DEC+48)
1570 IF DEC>9 THEN HEX$=HEX$+CHR$(DEC+55)
1580 RETURN
1600 REM HEX TO DEC CONVERT
1610 FOR I1=4 TO 1 STEP -1\W$=MID$(HEX$,I1,1)
1620 IF W$<="9" THEN D1=VAL(W$) ELSE D1=ASCII(W$)-55
1630 IF I1=4 THEN DN=D1
1640 IF I1=3 THEN DN=DN+(D1*16)
1650 IF I1=2 THEN DN=DN+(D1*256)
1660 IF I1=1 THEN DN=DN+(D1*4096)
1670 NEXT I1\DEC$=NUM$(DN)
1680 RETURN
```

5. LOAD2.BAS

```
SET NO DOUBLE
2000 PROGRAM LOAD2(HOME$,INVERSE$,NORMAL$,FLASH$,CENTER$)
2001 REM 19 DEC 85
2005 CENTER1$=CHR$(27)+"[13;35H"
2010 DIM #4,DA$(1400,1)\ PRINT HOME$+INVERSE$+"*** DUMP DATA ***"+NORMAL$
2015 OPEN "INTERIM" AS FILE #4, VIRTUAL
2016 PRINT CENTER$;INVERSE$;"CLEARING INTERIM STORAGE";NORMAL$
2018 FOR N1=0 TO 1400\DA$(N1,0)=""\DA$(N1,1)=""\ NEXT N1
2020  OPEN "XK:" AS FILE #2%\ REM COMMUNICATIONS PORT
2030 PRINT \ PRINT INVERSE$;"SWITCH INSTRUMENT CONTROL TO RESET THEN PRESS ";
2040 PRINT "RETURN";NORMAL$\ INPUT A$
2050 LINPUT #2,A$\ IF LEN(A$)>0 THEN 2050\ REM CLEAR COMM PORT
2055 IF LEN(A$)>0 THEN 2050
2060 PRINT \ PRINT "SWITCH INSTRUMENT CONTROL TO ";FLASH$;"RUN";NORMAL$;" NOW"
2070  LINPUT #2%,A$\ IF LEN(A$)<1 THEN D%=D%+1\ GOTO 2100 ELSE DA$(N,0)=DA$(N,0)+A$
2080 IF N=0 THEN PRINT HOME$;CENTER$;FLASH$;"DUMPING DATA";NORMAL$
2090  IF LEN(DA$(N,0))<6 THEN 2070 ELSE N=N+1\ PRINT CENTER1$;N\D%=0\ GOTO 2070
2100  IF D%<800 THEN 2070
2110 CLOSE #2%
2120 PRINT HOME$+CENTER$+INVERSE$+"FORMATTING DATA"+NORMAL$\ GOSUB 2290
2130  CLOSE #2%
2140  PRINT HOME$
2150  FOR N1=0 TO N\ PRINT DA$(N1,1),\ NEXT N1\ PRINT
2160  PRINT FLASH$+"DUMP TO DISK? (N OR RETURN)"+NORMAL$;\ INPUT A$
2170  IF A$="N" THEN 2280
2180  PRINT "FILENAME ";\ INPUT NA$
2190  PRINT HOME$+CENTER$+INVERSE$+"WRITING ";NA$+NORMAL$
2200  OPEN NA$ FOR OUTPUT AS FILE #3%
2210  PRINT #3%,N
2220  FOR N1=0 TO N
2230  PRINT #3%,DA$(N1,1)
2240  NEXT N1
2250  CLOSE #3%
2260  PRINT HOME$+CENTER$+FLASH$+"FILE ";NA$;" STORED"+NORMAL$\ PRINT
2270  PRINT "PRESS RETURN FOR MENU";\ INPUT A$
2275 CLOSE
2280 CHAIN "LOADER.BAS"
2290 REM RECONFIGURE DATA FROM D$ TO DA$
2295 D$=""\N3=0
2300 FOR N1=0 TO N-1
2320 FOR N2=1 TO LEN(DA$(N1,0))\A$=MID$(DA$(N1,0),N2,1)
2330 IF ASCII(A$)>47 AND ASCII(A$)<90 THEN D$=D$+A$
2335 IF LEN(D$)=4 THEN DA$(N3,1)=D$\N3=N3+1\D$=""\ PRINT CENTER1$;N3
2340 NEXT N2\ NEXT N1
2350 N=N3-1\ RETURN
```

6. LOAD3.BAS

```
SET NO DOUBLE
3000 PROGRAM LOAD3(HOME$,INVERSE$,NORMAL$,FLASH$,CENTER$)
3001 REM 19 DEC 85
3005  DIM #4,DA$(1400,1)\ PRINT HOME$+INVERSE$+"*** DISPLAY FILE DATA ***"+NORMAL$
3006  OPEN "INTERIM" AS FILE #4,VIRTUAL
3007  PRINT CENTER$;INVERSE$;"CLEARING INTERIM STORAGE";NORMAL$
3008  FOR N=0 TO 1400\DA$(N,0)=""\NEXT N
3010  GOSUB 9000
3020  FOR N1=0 TO N\ PRINT USING "####  'LLL ",N1,DA$(N1,0);
3030  IF N1-INT(N1/84)*84=83 THEN 3035 ELSE 3040
3035 PRINT \ PRINT "PRESS RETURN TO CONTINUE";\ INPUT A$\ PRINT HOME$
3040  NEXT N1\ PRINT \ PRINT "DISPLAY DATA AGAIN? (Y OR RETURN)";\ INPUT A$
3045  IF A$="Y" THEN PRINT HOME$\ GOTO 3020
3048  CLOSE
3050  CHAIN "LOADER.BAS"
9000  PRINT \ PRINT "FILENAME ";\ INPUT NA$
9005  PRINT HOME$+CENTER$+INVERSE$+"READING ";NA$+NORMAL$
9010  OPEN NA$ FOR INPUT AS FILE #2%
9030  INPUT #2%,N\ REM  NUMBER OF RECORDS IN FILE
9040  FOR N1=0 TO N
9050  INPUT #2%,DA$(N1,0)
9060  NEXT N1
9070  CLOSE #2%
9075  PRINT HOME$
9080  RETURN
```

7. LOAD3A.BAS

```
SET NO DOUBLE
4000 PROGRAM LOAD3A(HOME$,INVERSE$,NORMAL$,FLASH$,CENTER$)
4001 REM 19 DEC 85
4002 DIM #4,DA$(1400,1)
4003 OPEN "INTERIM" AS FILE #4, VIRTUAL
4005  PRINT HOME$+INVERSE$+"**** PRINT FILE DATA ***"+NORMAL$
4006 PRINT CENTER$;INVERSE$;"CLEARING INTERIM STORAGE";NORMAL$
4007 FOR N=0 TO 1400\DA$(N,0)=""\ NEXT N
4010  GOSUB 9000
4015 PRINT HOME$;CENTER$;INVERSE$;"PRINTING ";NA$;NORMAL$
4020  OPEN "LP:" FOR OUTPUT AS FILE #2%\ REM SET LINE PRINTER OUTPUT
4035  PRINT #2%,NA$\ PRINT #2%
4040  FOR N1=0 TO N STEP 6
4050  FOR N2=0 TO 5
4055  PRINT #2% USING "#### 'LLLL  ",INT(N1+N2),DA$(N1+N2,0);
4060  NEXT N2\ PRINT #2%\ NEXT N1
4070  CLOSE
4080 CHAIN "LOADER.BAS"
9000  PRINT \ PRINT "FILENAME ";\ INPUT NA$
9005  PRINT HOME$+CENTER$+INVERSE$+"READING ";NA$+NORMAL$
9010  OPEN NA$ FOR INPUT AS FILE #2%
9030  INPUT #2%,N\ REM  NUMBER OF RECORDS IN FILE
9040  FOR N1=0 TO N
9050  INPUT #2%,DA$(N1,0)
9060  NEXT N1
9070  CLOSE #2%
9075  PRINT HOME$
9080  RETURN
```

8. LOAD4.BAS

```
5000 PROGRAM LOAD4(HOME$,INVERSE$,NORMAL$,FLASH$,CENTER$)
5001 REM 19 DEC 85
5002 DECLARE DOUBLE A,B,R1,R2,AL,BE,GA,TM,LR\DIM #4,DA$(1400,1)
5003 OPEN "INTERIM" AS FILE #4, VIRTUAL
5005  PRINT HOME$+INVERSE$+"*** CONVERT & PRINT FILE DATA ***"+NORMAL$
5007 PRINT CENTER$;INVERSE$;"CLEARING INTER[illegible]
5008 FOR N=0 TO 1400\DA$(N,0)=""\ NEXT N
5010  PRINT "INSTRUMENT NUMBER? ";\ INPUT A$\I=VAL(A$)
5015  RESTORE
5020  FOR N=0 TO I\ READ A,B,R1,AL,BE,GA\ NEXT N
5030  PRINT "CRUISE NUMBER? ";\ INPUT CR$\ PRINT "DATE? ";\ INPUT DA$
5040  GOSUB 9000\N2=N
5045 OPEN NA$+".TXT" FOR OUTPUT AS FILE #1%
5050 PRINT HOME$;CENTER$;INVERSE$;"CONVERTING & PRINTING";NORMAL$
5060  DATA  .14797245,.55253495,8715,8.9910055E-4,2.4362808E-4,1.2626278E-7
5062  DATA  .14797245,.55253495,8715,8.9910055E-4,2.4362808E-4,1.2626278E-7
5064  DATA  .14797245,.55253495,8715,8.9910055E-4,2.4362808E-4,1.2626278E-7
5066  DATA  .14797245,.55253495,8715,8.9910055E-4,2.4362808E-4,1.2626278E-7
5068  DATA  .14797245,.55253495,8715,8.9910055E-4,2.4362808E-4,1.2626278E-7
5070  DATA  .14797245,.55253495,8715,8.74722104E-4,2.47238861E-4,1.16103772E-7
5072  DATA  .14797245,.55253495,8715,8.84648075E-4,2.45607753E-4,1.21657048E-7
5074  DATA  .14797245,.55253495,8715,8.78525336E-4,2.46894380E-4,1.17247016E-7
5076  DATA  .14797245,.55253495,8715,8.83512916E-4,2.46027504E-4,1.17903032E-7
5078  DATA  .14797245,.55253495,8715,8.50827418E-4,2.50936456E-4,1.03111000E-7
5080  DATA  .14797245,.55253495,8715,8.78246674E-4,2.46585729E-4,1.17843303E-7
5100  OPEN "LP:" FOR OUTPUT AS FILE #2%\ REM SET UP LINE PRINTER
5105 L%=0\ REM LINE COUNTER
5110  PRINT #2%,"CRUISE # ";CR$;"   DATE ";DA$\L%=L%+1
5115  PRINT #1%,"CRUISE # ";CR$;"   DATE ";DA$
5120  PRINT #2%,"INSTRUMENT # ";I\L%=L%+1
5125  PRINT #1%,"INSTRUMENT # ";I
5130  REM PRINT  CHR$ (27); CHR$ (58);
5140  PRINT #2%,NA$
5145  PRINT #1%,NA$+".DAT"
5150  PRINT #2%\ PRINT #2%," REC  COUNT   RESISTANCE      TEMP"\L%=L%+3
5155  PRINT #1%\ PRINT #1%," REC  COUNT   RESISTANCE      TEMP"
5160  FOR I=0 TO N2
5170  FOR I1=4 TO 1 STEP -1\C$=MID$(DA$(I,0),I1,1)
5180  IF I1=1 THEN IF ((C$>"3") AND (C$<"8")) OR (C$>"B") THEN DN=9999
5190  IF I1=1 THEN 5260
5200  IF C$<="9" THEN D1=VAL(C$)\ GOTO 5220
5210  D1=ASCII(C$)-55
5220  IF I1=4 THEN DN=D1
5230  IF I1=3 THEN DN=DN+(D1*16)
5240  IF I1=2 THEN DN=DN+(D1*256)
5250  GOTO 5270
5260  IF C$<"8" THEN DN=-DN
5270  NEXT I1
5280  PRINT #2% USING "####",I;
5285  PRINT #1% USING "####",I;
5290  IF MID$(DA$(I,0),1,4)="STOP" THEN PRINT #2%,"   STOP";\ PRINT #1%,"   STOP";\GOTO 5360
5300  IF DN=9999 THEN PRINT #2%,"   OVER";\ PRINT #1%,"   OVER";\GOTO 5360
5310  PRINT #2% USING "#######",DN;\PRINT #1% USING "#######",DN;
5320  R2=R1*(1/(1/(((A*DN)/2048)+B)-1))\ PRINT #2% USING "#########.##",R2;\PRINT #1% USING "#########.##",R2;
5330  LR=LOG(R2)\TM=(1/(AL+LR*BE+LR^3*GA)-273.15)
5340  PRINT #2% USING "  #####.###",TM\L%=L%+1\PRINT #1% USING "  #####.###",TM
```

```
5345 IF L%=60 THEN L%=0\ PRINT #2%,CHR$(12)
5360  NEXT I
5370  PRINT #2%,CHR$(12)
5380  CLOSE #2%\CLOSE #1%
5385 CLOSE
5390 CHAIN "LOADER.BAS"
9000  PRINT \ PRINT "FILENAME ";\ INPUT NA$
9005  PRINT HOME$+CENTER$+INVERSE$+"READING ";NA$+NORMAL$
9010  OPEN NA$ FOR INPUT AS FILE #2%
9030  INPUT #2%,N\ REM  NUMBER OF RECORDS IN FILE
9040  FOR N1=0 TO N
9050  INPUT #2%,DA$(N1,0)
9060  NEXT N1
9070  CLOSE #2%
9075  PRINT HOME$
9080  RETURN
```

APPENDIX H. COMPUTER PROGRAMS TO REDUCE HPC TEMPERATURE DATA

LANGUAGE: FORTRAN-4

MACHINE: VAX 11/780 (Virtual Memory System)

INTRODUCTION - Two programs (DECAY1, FITTING1) have been developed to extrapolate temperature measured over time after penetration of a core barrel (HPC) into bottom sediments. These are presently written as separate programs, although they could be combined as subroutines called by a main program.

REFERENCE: K. Horai, A theory of processing downhole temperature data taken by the hydraulic piston corer (HPC) of the DSDP, ms. in preparation.

DECAY1 - Computes theoretical cooling temperature of an idealized core barrel in sediment, for the normalized initial temperatures of 1°C for the HPC, and 0°C for the sediment. The core barrel is assumed to be a perfect conductor (uniform temperature) although with finite heat capacity, with 2-dimensional (radial) conduction to the sediment.

Instructions for running the program from a computer terminal

1. You will be asked by the terminal for values of a and b; the inner and outer radii, respectively, of the HPC. The values of a and b (units of meters) are typed in from the terminal.

2. Give a 6-digit file name to identify the cooling curve. If you give uvwxyz, the file name to be created will be Tuvwxyz.DAT.

3. Then you will be asked for the thermal conductivity of the sediment. Type in its value (units of W/mK).

4. Then you will be asked for the time interval between data points. Check the temperature record to be compared with the theory, and give it in the units of seconds.

5. The program computes 3500 terms of the integrand given in the theory (these are displayed on the terminal as computations progress) and 200 cooling temperatures at the specified time interval by numerically integrating the integrands, multiplied by an exponentially decaying term as given in the theory. The results are stored in the file.

FITTING1 - Compares the theoretical cooling curve with the temperature record to determine the optimal fit by least-squares analysis.

Instructions to run the program:

1. You will be asked for the 6-digit thermal decay file name. Type in from the terminal keyboard Tuvwxyz.DAT file.

2. Then you will be asked for a 6-digit file name to specify the corresponding data. Type in from the terminal <u>Dabcdef.DAT</u> to open this data file.

3. Type in the number (IO) of "bad" (poorly fitting) data points (the first part of the cooling curve) that are not used in the analysis. Sometimes this must be done iteratively (trial and error) for noisy or otherwise uncertain data.

4. The printout will give the residual fit for various shift (NO) in origin of data points vs. theoretical curve. This is done to determine an effective origin time for uncertain penetrations. The data and the theoretical cooling temperatures will be printed for the value of NO that gives the minimum residual.

DOCUMENT LIBRARY

April 9, 1985

DISTRIBUTION LIST FOR TECHNICAL REPORT EXCHANGE

Institute of Marine Sciences Library
University of Alaska
O'Neill Building
905 Koyukuk Ave., North
Fairbanks, AK

Attn: Stella Sanchez-Wade
Documents Section
Scripps Institution of Oceanography
Library, Mail Code C-075C
La Jolla, CA 92093

Hancock Library of Biology & Oceanography
Alan Hancock Laboratory
University of Southern California
University Park
Los Angeles, CA 90089-0371

Gifts & Exchanges
Library
Bedford Institute of Oceanography
P.O. Box 1006
Dartmouth, NS, B2Y 4A2, CANADA

Office of the International
Ice Patrol
c/o Coast Guard R & D Center
Avery Point
Groton, CT 06340

Library
Physical Oceanographic Laboratory
Nova University
8000 N. Ocean Drive
Dania, FL 33304

NOAA/EDIS Miami Library Center
4301 Rickenbacker Causeway
Miami, FL 33149

Library
Skidaway Institute of Oceanography
P.O. Box 13687
Savannah, GA 31416

Institute of Geophysics
University of Hawaii
Library Room 252
2525 Correa Road
Honolulu, HI 96822

Library
Chesapeake Bay Institute
4800 Atwell Road
Shady Side, MD 20876

MIT Libraries
Serial Journal Room 14E-210
Cambridge, MA 02139

Director, Ralph M. Parsons Laboratory
Room 48-311
MIT
Cambridge, MA 02139

Marine Resources Information Center
Bldg. E38-320
MIT
Cambridge, MA 02139

Library
Lamont-Doherty Geological Observatory
Colombia University
Palisades, NY 10964

Library
Serials Department
Oregon State University
Corvallis, OR 97331

Pell Marine Science Library
University of Rhode Island
Narragansett Bay Campus
Narragansett, RI 02882

Working Collection
Texas A&M University
Dept. of Oceanography
College Station, TX 77843

Library
Virginia Institute of Marine Science
Gloucester Point, VA 23062

Fisheries-Oceanography Library
151 Oceanography Teaching Bldg.
University of Washington
Seattle, WA 98195

Library
R.S.M.A.S.
University of Miami
4600 Rickenbacker Causeway
Miami, FL 33149

0272-101

REPORT DOCUMENTATION PAGE	1. REPORT NO. WHOI-86-3	2.	3. Recipient's Accession No.

4. Title and Subtitle **A Miniature Deep Sea Temperature Data Recorder: Design, Construction, and Use**	5. Report Date January 1986
	6.
7. Author(s) R. Koehler and R. P. von Herzen	8. Performing Organization Rept. No. WHOI-86-3
9. Performing Organization Name and Address Woods Hole Oceanographic Institution Woods Hole, Massachusetts 02543	10. Project/Task/Work Unit No.
	11. Contract(C) or Grant(G) No. (C) OCE 82-14658; (G) OCE 83-00073
12. Sponsoring Organization Name and Address National Science Foundation	13. Type of Report & Period Covered Technical
	14.

15. Supplementary Notes

This report should be cited as: Woods Hole Oceanog. Inst. Tech. Rept. WHOI-86-3.

16. Abstract (Limit: 200 words)

A miniature temperature recorder has been developed to be used with the hydraulic piston sediment corer (HPC) on the Deep Sea Drilling Project (DSDP). The instrumentation fits into pressure-sealed slots in the wall of the HPC, allowing temperature measurements to be made simultaneously with coring operations. Temperatures from -2 to 70°C are measured to a resolution of about 0.01°C. Up to 1300 13-bit measurements are recorded in random access memory (RAM), at a sampling rate ranging between 0.1 s to over 100 min., as specified by the operator in a program loaded into a microprocessor of the instrument. During recording the instrumentation uses about 3.5 mamp at 7.5 volts, which can be supplied for about 20 hours of operation by a custom-made pack of silver-oxide batteries. The corer is normally left motionless in the sediment for about 10 min. to allow extrapolation of the measured temperatures to equilibrium *in situ* temperature. Examples of data from DSDP Leg 86 are given.

17. Document Analysis a. Descriptors

1. miniature temperature recorder
2. heat flow
3. drilling instrumentation

b. Identifiers/Open-Ended Terms

c. COSATI Field/Group

18. Availability Statement:	19. Security Class (This Report)	21. No. of Pages
Approved for publication; distribution unlimited.	UNCLASSIFIED	59
	20. Security Class (This Page)	22. Price

ANSI-Z39.18) See Instructions on Reverse OPTIONAL FORM 272 (4-77) (Formerly NTIS-35) Department of Commerce